1990

A caption-based natural-language interface handling descriptive captions for a multimedia database system

Dulle, John David

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/30639

②

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A CAPTION-BASED NATURAL-LANGUAGE INTERFACE HANDLING DESCRIPTIVE CAPTIONS FOR A MULTIMEDIA DATABASE SYSTEM

by

John D. Dulle

June 1990

Thesis Advisors: Vincent Y. Lum
Neil C. Rowe

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Computer Science Dept. Naval Postgraduate School — 6b. OFFICE SYMBOL (if applicable) 52	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
A Caption-Based Natural-Language Interface Handling Descriptive Captions For A Multimedia DataBase System

12. PERSONAL AUTHOR(S)
Dulle, John D.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Master's Thesis	FROM ____ TO ____	June 1990	145

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the authors and dc not reflect the official policy or position of the Department of Defense or the U.S. Government

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
Natural Language Processing, Multimedia Database System, Natural Language Interface, Descriptive Captions

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

This research examined the grammar structure of descriptive English captions on multimedia data. The research was composed of three phases. The first phase was to investigate the grammar structure of example descriptive captions from a variety of subject domains. The second phase was to develop a set of domain-independent binary grammar rules to be used in the Caption-Based Interface (CBI) which is a natural language interface for the Multimedia Database System. The third phase of the research was to implement and test the grammar rules in the CBI. The program was implemented in C-Prolog on a Sun SPARC workstation. The testing phase also includes timing and memory comparisons between C-Prolog an interpretive programming language and a compiled version of the code using Quintus Prolog. This thesis was able to show that the grammar rules that were developed could correctly identify their intended structures. Another accomplishment of this thesis was to demonstrate that the CBI could parse 25 out of 30 example captions and that it could correctly handle some semantic interpretations of the parsed captions.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [✓ UNCLASSIFIED/UNLIMITED] ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL Lum, Vincent Y., and Rowe, Neil C.	22b. TELEPHONE (Include Area Code) (408) 646-3091/2462 — 22c. OFFICE SYMBOL 52Lu/52Rp

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

A CAPTION-BASED NATURAL-LANGUAGE INTERFACE HANDLING DESCRIPTIVE CAPTIONS FOR A MULTIMEDIA DATABASE SYSTEM

by

John David Dulle
Captain, United States Marine Corps
B.B.A., University of Oklahoma

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1990

Author: [redacted]
John David Dulle

Approved By: [redacted]
Vincent Y. Lum, Thesis Advisor

[redacted]
Neil C. Rowe, Thesis Advisor

[redacted]
Robert B. McGhee, Chairman,
Department of Computer Science

ABSTRACT

This research examined the grammar structure of descriptive English captions on multimedia data. The research was composed of three phases. The first phase was to investigate the grammar structure of example descriptive captions from a variety of subject domains. The second phase was to develop a set of domain-independent binary grammar rules to be used in the Caption-Based Interface (CBI) which is a natural language interface for the Multimedia Database System. The third phase of the research was to implement and test the grammar rules in the CBI. The program was implemented in C-Prolog on a Sun SPARC workstation. The testing phase also includes timing and memory comparisons between C-Prolog an interpretive programming language and a compiled version of the code using Quintus Prolog. This thesis was able to show that the grammar rules that were developed could correctly identify their intended structures. Another accomplishment of this thesis was to demonstrate that the CBI could parse 25 out of 30 example captions and that it could correctly handle some semantic interpretations of the parsed captions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENTS

I thank Dr. Vincent Lum and Dr. Neil Rowe for all of their support and guidance throughout the development of this thesis. I also thank Dr. Bernhard Holtkamp and Gene Guglielm for being sounding boards when I needed them.

Most importantly I thank my lovely wife, Laurie, and my two daughters Sarah and Kristi, for their patience and support through the writing of this thesis.

I. INTRODUCTION

A. THE PROBLEM

In the military as well as the private sector there is always a demand to analyze much more information than a human analyst can process in a relative short time frame if at all. In general the analyst is required to interpolate information that comes from numerous sources and in various forms. These forms fall into basically three types: sounds, images, and text. In military applications, the intelligence officer might receive as data a string of sounds such as sonar sounds, aerial photographs of ships at sea, and a set of operational orders. Each of these data items contain a tremendous amount of information.

Take for example a picture of a ship alongside a dock. This picture contains a lot of information that is both explicit and implicit. The explicit information would be such things as the color of the ship, the type of ship, the ship's hull number, what type of dock, etc. The implicit information would be such items as where is the dock located what time of year the picture was taken, etc. All of this information is valuable to the military analyst in various degrees and what is not valuable to one analyst might be to another. The value of the data is in how the data is related to other data items that the analyst has.

The use of computers with database management systems has provided the analyst the ability to store and manipulate numbers and keywords in such a manner that makes the information useful to the analyst and decision makers. The analyst has the ability to do content searches in order to acquire related information. This technology is well understood. If we go back to our example given above though a picture contains a vast amount of data. The phrase "a picture is worth a thousand words" takes on real significance at this point. The information that is key to the analyst can be both implicit and explicit. This now brings us to the problem of how should the picture be stored and what has to be done to retrieve the picture.

B. A POSSIBLE SOLUTION

The abilities to physically store images, sounds and text in computers are tasks that have been accomplished in the recent years. At the present time in the experimental Multimedia Database System (MDBMS) at the Naval Postgraduate School we have the ability to physically store images [Ref. 1] and sounds [Ref.2]. The next problem to be conquered is how to store information about the picture or string of sounds and how do we retrieve such information. We have chosen to store the various types of data through the use of descriptive captions [Ref. 3]. These captions can be thought of in the same manner as those one would see in a book or magazine. It is our proposal that the descriptive captions for the MDBMS should use the English natural language.

The field of natural language processing has been an area of interest for 30 plus years to a number of disciplines such as Computer Science, Linguistics, and Education and to date has not been completely solved. There have been many attempts to solve this problem over the years with various algorithms and computer languages. We are not naive enough to believe that our program will in fact be a "complete" natural language processor. It is our hypothesis that descriptive captions are only a subset of the natural language and therefore a doable project. Descriptive captions should work as a natural interface for the multimedia database system.

C. DESCRIPTION OF THESIS

This thesis is the preliminary work for the idea of a natural language processor for descriptive captions. The work that is described here was in three phases. The first phase was to determine the grammatical structure of captions from a random selection of books that contained pictures with captions. The second phase was to investigate the use of the computer to recognize the syntactic structure of the captions. The third phase was to start work on the correct semantic interpretations of given phrases.

The remaining chapters of this thesis provides some background to the problem and provides our method of approach to the problem. Chapter II provides a couple of examples of attempted solutions to natural language processing. Chapter III is an examination of our approach and the assumptions we made. Chapter IV is a detailed description of our program. Chapter V discusses the results of the output from the program. Finally, some conclusion and areas of follow-on research are presented in Chapter VI.

II. BACKGROUND

A. INTRODUCTION

Most work in the natural language field has focused on incorporating the full functionality of the English Language. Natural language processors usually contain three components: a parser, a semantic interpreter, and a contextual interpreter. This thesis deals primarily with the parser and to a lesser degree the semantic interpreter.

The role of the parser is to pick apart the structure (syntax) of an English sentence, using the information provided by the language's grammar, in order to help determine the sentence's meaning [Ref. 4:p. 229]. The following sections of this chapter provide an overview of the different types of grammars and parsing techniques that are currently being used in natural language processing systems. A description of a natural language interface system is also provided to illustrate some of these concepts.

B. TYPES OF GRAMMARS

A grammar is defined as the formal specification of the structures allowable in a language [Ref. 5:p. 41]. This is the scheme for specifying the sentences that are allowed in the language, indicating the rules for combining words into well-formed phrases and clauses [Ref. 4:p. 229]. A grammar is a collection of rules that can be used in a systematic way to generate the sentences of a language by putting strong constraints on the patterns that are used in a language [Ref. 6:p. 72]. Some grammars that have been developed are context-free, augmented phrase structure, recursive transition network, and augmented transitions network.

1. Context-free Grammars

A context-free grammar consists of a set of rules, each representing a labeled pattern to be matched against a sequence of constituents [Ref. 6:p. 82]. An example of a simple context free grammar is illustrated in Figure 1 (see Appendix C for a list of the abbreviations used). In this grammar the symbols on the left side of the arrow represent the pattern name while the symbols on the right represent the syntactic rules that make up the pattern. The symbols d(eterminer), adj(ective), noun, and verb located on the right hand side represent terminals of the grammar. These terminals represent word categories which are normally found in the lexicon.

```
snt --> np  vp
np --> d  ng
ng --> adj  noun
ng --> noun
vp --> vg
vp --> vg  np
vg --> verb

d --> the
adj --> tall
adj --> big
noun --> Marine
noun --> ship
verb --> fought
```

Figure 1 -Simplified Context-free Grammar

In a context-free grammar the left-hand side must only have a single nonterminal symbol. An important property to note is that every derivation can be represented in a tree structure. Using the grammar of Figure 1 to illustrate this point, Figure 2 is a derivation of the sentence "The tall Marine fought the enemy".

The advantage of context-free grammars is that they are restrictive enough to allow an efficient parser to analyze sentences while at the same time they are able to handle most structures in natural languages [Ref. 5:p. 41]. A context-free grammar provides a simple way of describing the structures of a language and of setting up a correspondence between the knowledge structures, the structures generated in producing or recognizing a sentence, and the process of recognition and production [Ref. 6:p. 72].

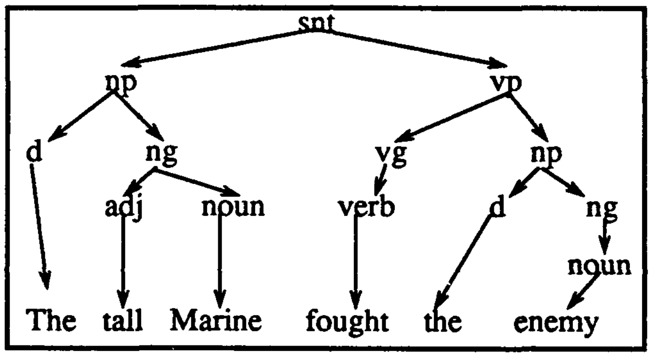

Figure 2 -Tree Structure Derivation

2. Augmented Phrase Structure Grammar

An augmented phrase structure grammar is an extension of the context-free grammar. The context-free grammar is augmented with role and feature registers, conditions, and actions. Roles and features are given to each syntactic category and a set of conditions and actions are attached to each rule. The conditions use the information in the role and features to prevent a rule from being applied if a test fails. The actions are used to set the role and features that are associated with the nodes of the constructed parse tree.[Ref. 6:p. 377]

3. Recursive Transition Networks

A recursive transition network (RTN) grammar consists of a set of labeled networks instead of a set of rules as with the context-free grammar described above. Figure 3 provides an example of a RTN using the context-free grammar of Figure 1. The basic structure of each network consists of a set of states, connected by labeled arcs. The arcs that are labeled with syntactic categories i.e., np, vp, etc., are also the label for a network. Each arc represents a transition between two states and is transversed by matching a sequence of input symbol the different networks. [Ref. 6:p. 197]

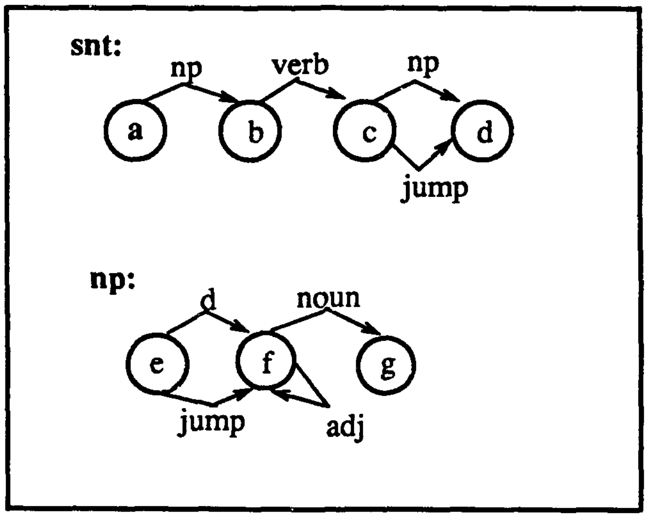

Figure 3 -Recursive Transition Network

4. Augmented Transition Networks

An augmented transition network (ATN) is similar to a recursive transition network as described above with some minor modifications. The major differences between these two approaches is that the nodes of an ATN are augmented with conditions and actions that are associated with the arcs of the network. The conditions restrict the circumstances under which and arc can be taken while actions perform feature-marking and structure-building operations. [Ref. 6:p. 204]

The basic structure of an ATN is also similar to the RTN but it has some additional features. First the ATN uses a set of registers that store information about partially derived trees and jumps between different subnetworks. The second difference is that the arcs, in addition to being labeled, have tests that are associated with them. These tests must be satisfied before the arc can be taken. The last major difference is that certain actions may be attached to an arc that will be performed whenever an arc is taken.

C. PARSING METHODS

Parsing methods can be divided into three different basic categories: top-down, bottom-up, and mixed-mode. Different grammars with various parsing algorithms lend themselves to the use of one of these types.

1. Top-Down

A top-down parser begins with the rules for the top level structure and looks at its constituents. It then looks at the rules for the constituents by breaking them down until the terminal levels of each applicable rule are reached and a complete sentence structure is formed. This structure is compared to the input data and if it matches the parse is complete; otherwise the parser starts back at the top level and generates another sentence structure. [Ref. 4:p. 259]

The primary advantage of the top-down method is that word categories that are not in their proper position in the sentence will not be considered. For example, using in the grammar in Figure 1 the sentence "The go ship" would not parse because the only thing that can follow the determiner "the" is a ng. A disadvantage to the top-down parsing though, is that the parser might take a long time to get to the actual words because the same rule may have to be considered many times while the parser is searching of a solution. [Ref. 5:p. 66]

2. Bottom-Up

A bottom-up parser begins with the words and looks for rules whose right-hand sides match. It then tries to combine these rules with each other and the remaining words into larger constituents, and proceeds up the structure tree until it is able to combine constituents covering the entire input into a single structure labeled with the distinguished symbol. [Ref. 6:p. 90]

The primary disadvantage of this method is that the parser must consider all senses of each word. This leads to grammar structures that should never be considered.

3. Mixed-Mode

The mixed-mode method tries to combine the advantages of the top-down and the bottom-up methods. As seen in the two methods above it seems pretty clear that there needs to be a parsing method that would only consider words that are from the proper category while at the same time only construct the rules once. One way of accomplishing this is by modifying the top-down parser by adding a cache to it. The purpose of the cache is to hold the possible lexicon values for the word in the sentence. Then by adding a condition to the rules, as to what is allowed, those rules that do not have a matching condition will not be considered.

D. A REPRESENTATIVE SYSTEM

Numerous applications require natural language interfaces. This has resulted in many attempts to attack/solve the natural language problem. One representative system is TEAM (Transportable English database Access Medium) which will be discussed next.

1. Overview

TEAM is a transportable natural language interface system. TEAM was developed to test the feasibility of providing a natural language interface for databases. The key premise is that the system could be transported from one database to another. Another key concern was that when moving form one database to another the database expert and users of the database would not be required to have special knowledge about natural language processing.[Ref. 7:p. 175]

The system is used in two major ways: acquisition and question-answering. The acquisition mode is an interactive process in which the database expert provides information about the files and fields in the database, the conceptual concept, and the words and phrases used to refer to the concepts. The question-answering system provides the logical representations of the meanings from the natural language expressions that the user enters and then translates these representations into statements of the database query language. The question-answering system is encapsulated in DIALOGIC and it is the portion of TEAM that this thesis addresses.[Ref. 7:p. 177]

2. DIALOGIC

DIALOGIC performs the task of transforming a query's literal meaning into a database query. In order to make the transformation, DIALOGIC must first construct a logical representation of the query's meaning. This is accomplished with the DIAGRAM grammar, the DIAMOND parser, and the lexicon.

a. The DIAGRAM Grammar

The DIAGRAM grammar is an augmented phrase structure grammar with rules that cover a broad spectrum of constructions, including all common sentence types, complex auxiliaries and modals, complex noun phrases, all the common quantifiers, comparative and measure expressions, relative clauses and subordinate clauses, adverbial modifiers and a limited range of conjunctions. [Ref. 7:p. 185]

b. The DIAMOND Parser

The Diamond parser is a bottom-up parser and is designed to work closely with the augmented phrase structure grammar of DIAGRAM. As a bottom-up parser it tries to put together the pieces of the grammar as it comes to them. In order to do this it uses an active chart algorithm (see Ref. 6:pp 120-126 for further information) that keeps track of what has already been parsed. Using conditions and actions the parser weights the grammar rules constituents so that while the structure is being built it will build along the most likely path.[Ref. 6:pp. 381-382]

c. The Lexicon

The lexicon provides information about each word that is to be used in the database. The lexicon contains two different classes of words: the closed class and the open class. The closed class are those words that will be common from domain to domain. These are syntactic classes such as determiners, pronouns, and conjunctions. Open class words on the other hand are domain-dependent and are syntactic classes such as nouns, verbs and adjectives. For each sense of the word the lexicon maintains syntactic and semantic information.[Ref. 7:p. 178]

III. APPLICATION DESCRIPTION

A. ASSUMPTIONS

1. General

When we started out this project our primary assumption was that captions were a subset of the natural language. It was our premise that we would find that captions were primarily fairly simple noun phrases and simple sentence constructions.

2. Caption Selection

We attempted to prove our premise by looking at a number of captions from different domains. After determining that the style of captions was not domain dependent we concluded that the grammar we would develop would not be domain dependent also. Therefore, before selecting any of the captions we decided to focus on one particular domain. The reason for this was to keep our vocabulary to a manageable testing size.

We chose photographs from World War II as our domain. The domain was further constrained as follows: photographs had to be of action in the Pacific; the photographs had to be of ships; and the photographs had to be taken from the air. With this criteria we selected 13 captions from pictorial history books (Ref. 8-11) that met these conditions. Appendix A provides a full listing of the selected captions.

These caption consisted of a total of 30 sentences. Next we manually parsed each sentence, the results of which is in Appendix D, and treated each sentence as a caption. With this information we developed the grammar that is used in our parser.

3. Caption Modification

After some initial analysis of the chosen 13 captions we noticed that there was a need to make some modifications in order to simplify the manual and automatic analysis. The first major modification was to treat each sentence as a separate caption. This gave a total of 30 different sentence to analyze.

The second modification was to the sentences. In an attempt to simplify the grammar rules and the parsing of the captions we modified some sentence punctuation and number representation. The type of punctuation modifications were the removal of quotation marks and the removal of periods in abbreviations, i.e. if the original caption had U.S. we changed it to US. The numerical modifications were to change any number over ten to Arabic numerals and to remove the comma for any number that was 1,000. Appendix B provides the modified captions and can be easily cross-referenced to Appendix A for a more detailed look at the modifications.

B. APPLICATION

The main focus of this thesis was to determine the grammar to be used by the Captioned-Based Interface and to implement those rules in the parser. The primary objective of the parser is transforming the natural language description into a set of predicates and literals that logically represent the description, but at the same time provide semantic information to the multimedia database for its query. As a by-product of this thesis, the efficiency and the capability of the parser was tested. In fact, as a result of this work the parser has been rewritten. Further details are given in the next chapter.

IV. IMPLEMENTATION

A. INTRODUCTION

The Caption-Based Interface (CBI) is a multi-facet project with a number of components to it. This project has been developed in a team approach here at the Naval Postgraduate School. The team consists of Dr. Neil Rowe, Dr. Vincent Lum, Dr. Bernhard Holtkamp, Gene Guglielm, and myself. During the implementation numerous decisions were made as to the capabilities, components, and the structure of the CBI, some of which were discussed in the previous chapter. The implementation of the Prolog code was also a multi-effort task and as I discuss the various components I will try to give credit were it is due. For a more detailed look of the code refer to Appendix G which provides the source code for the CBI and also shows the credits for the various procedures.

B. THE CAPTION-BASED INTERFACE COMPONENTS

The CBI is made up of seven main modules: the user interface module, the semantics module, the anaphoric module, the rules module, the parse module, the meanings module, and the dictionary (lexicon) module. The first three modules listed are not within the scope of this thesis and thus will only be briefly defined while the other four will be described in detail in the following sections.

The user interface module at the moment is in the form of a query. Once the user has activated the CBI, the system prompts the user for a partial caption and then the system will provide an interpretation. The semantics module analyzes predicates and resolves ambiguities. The anaphoric module contains the strategies for handling anaphoric references.

1. The Parse Module

The parser module is the top level of the parser and was developed by Dr. Rowe. It is designed to take two cached lists that are provided from the user interface module and output two lists. The first input list is a list of words to be parsed, i.e., the sentence that the user inputs. The second list is a list of possible parts of speech that each word in the sentence could have. The first output list contains the remaining list of words after the parse. The second output list contains the meaning list that is generated.

2. The Rules Module

The rules module constituted the bulk of the research for this thesis. In order to investigate our premise and assumptions, section III.A.1, we manually parsed each caption into a binary tree structure. The result of this is located in Appendix D.

Once we had the parsed captions we converted each binary branch into a grammar rule, i.e., if the parent of a branch was a **snt** and its children were **np** and **vp** this was converted to **snt --> np, vp.** We originally had 150 of these rules. After some analysis of these 150 rules it was seen that this number could be reduced because some of the rules could be defined in a more senior rule i.e., a **ng** is a subset of a **np.** This reduced the total required rules to 103 rules which are located in parse_rule procedure in Appendix F.

After some initial testing we found that the parser was extremely slow for a couple of reasons. The first was that the parser spent a lot of time going through rules that it did not need to go through because of their ordering in the code. The first fix was to order the rules in a top-down manner to prevent some of the backtracking. The first part of the ordering was to from a grammatical basis, i.e., a noun phrase must come before a verb phrase. The second part of the ordering was to order the rules that were used most often first, i.e., rules for prepositional phrases were placed ahead of rules for appositives. This improved the system slightly and the parser was able to parse some small uncomplicated sentence structures.

The other reason for the slowness of the parser was that the rules contained both left and right recursion i.e., a **np** could be defined as **np --> np, prtp**. The first attempt at fixing this problem was to remove all occurrences of this type of structure by breaking down complex grammar rules such as a **np** into a **basic_np, extended_np,** and **complex_np**. This approach ran into immediate problems because it increased the number of rules to almost 300, therefore this approach was abandoned.

In order to get increase the performance Dr. Rowe developed a look-ahead technique to be used in conjunction with the parse rules. The look-ahead code was applied in the parse module as described in section IV.B.1. Then by adding an additional list to the rules we turned the top-down parser into a mixed-mode parser, refer to section II.B.3. Figure 4 provides an example of this.

```
parse_rule(snt,[np,vp],
    [[noun,propnoun,pronoun,geo,month],
     [verb,tobe]]).
```

Figure 4 -Mixed-Mode Parse Rules

In the example in order for the **snt** rule to be evaluated, the word string that represents it must contain at least one of the following: a noun, a proper noun , a pronoun, a geographical location or a month. The cached word string must subsequently also contain either a verb or some form of the word "be".

Another performance enhacement developed by Dr. Rowe was a routine to cache previous parse successes and failures. The primary purpose of this was to avoid infinite loops.

The result of all of the modifications was that we ended up with the rules module being an augmented phrase structure grammar as described in section II.B.2 of this thesis.

3. The Dictionary Module

The dictionary module is responsible for determining the root of a word in order to return syntactic information along with a meaning list. In order for the module to determine the root of the word it checks whether nouns are plural or possessive, checks for the canonical form of verbs, and determines the verb that corresponds to a participle. This portion of the module was written by Dr. Rowe and modified by myself and Dr. Holtkamp. The dictionary module contains the parts of speech and the meaning list for each tense of the words used.

The dictionary was made up from words contained in the captions of Appendix B. This module was developed and enhanced by myself and Dr. Holtkamp. The words were placed in syntactic categories, i.e., nouns, verbs, adjectives, etc. There is a separate dictionary entry for each tense of the word. For example the word **land** is listed as an noun for the sentence "The Marines fought on **land**." and as a verb for the sentence "The Marines **landed** on the beach." At the present time there is a total of 634 dictionary entries.

4. The Meanings Module

The meanings module starts bridging the parser and the semantic interpreter of the CBI. This module was developed by Dr. Rowe and modified by Gene Guglielm and myself. The module combines the meaning lists of two parsed lists together to provide some semantic interpretation. This module sets up the relation that one part of speech has with another. This is done by manipulating the variables that have been assigned in the dictionary module. Figure 2 provides an example.

```
Input:
    The Marines fought the enemy.

Rules used:
snt --> snt, period [.]
    snt --> np, vp
        np  --> determiner [The], ng
            ng --> propnoun [Marines]
        vp --> vg, np
            vg --> verb [fought]
            np --> determiner [the], ng
                ng --> noun [enemy]

Output:
    My interpretation is:
       definite(c2)
       name(c2,Marines)
       military(c2)
       fight(c2,f2)
       subject(fight,c2)
       action(fight,d2)
       tense(d2,past)
       definite(f2)
       enemy(f2)
```

Figure 5 -Example Of Meaning List Variables

Using the example above we can see the relationship between the different variables. To illustrate how the combination takes place look at the section of output that starts with the word **fight**. "Fight" has two variables associated with it: **c2** and **f2**. The variable **c2** ties the verb phrase to the noun phase above while the **f2** ties the noun phase "the enemy" to the verb "fight" to form the verb phrase. The combining of variables must occur for every syntactic rule that is formed from two constituents.

C. SYSTEM REQUIREMENTS

The CBI was originally running on a VAX 11/785. But as the system grew it was temporarily moved to an ISI work station and then finally to a Sun Sparc work station. This final move provided much improved speed in parsing. The Sparc work stations in the Artificial Intelligence Lab at NPS have 6 megabytes of real memory and 7 megabytes of virtual memory and they run at 10 MIPS.

At the present time the CBI is implemented using the Prolog programming language. Prolog was chosen for its extremely powerful backtracking ability. The Interface will run on two different versions of Prolog: C-Prolog and Quintus Prolog. C-Prolog is an interpretive version of Prolog while Quintus is a compiled version. A comparison of the results between these two are described in Chapter V of this thesis.

D. PROGRAM INPUT REQUIREMENTS

The CBI was designed to handle captions that are typed in by the user of the system. At the present time captions can only be single noun phrases or single complete sentences, i.e., if a caption contains multiple sentences the user can only give the system one sentence at a time.

In order for the noun phrase to be recognized it must not end with a period. A sentence could be of varying degrees of complexity but must consist of at least a noun, then a verb and finally end with a period. Figure 3 illustrates some examples of valid inputs.

E. PROGRAM OUTPUT

The parser provides a grammatical interpretation for the sentence or noun phrase as output. Figure 4 shows a valid input and the system output.

NOUN PHRASES

An American war ship
A company of Marines on the ship
A Japanese ship under attack on January 15, 1943

SENTENCES

Marines fight.
The Marines landed on the beach.
The Marines wading ashore fought the Japanese.

Figure 6 -Examples of Valid Input

Input

The Morotai invaders met no resistance.

Output:

```
My interpretation is:
   definite(i2)
   name(i2,Morotai)
   place(i2)
   plural(i2)
   invader(i2)
   meet(i2,l2)
   subject(meet,i2)
   action(meet,j2)
   tense(j2,past)
   cardinality(l2,0)
   resistance(l2)
```

Figure 7 -Example of Output

V. TEST RESULTS

A. INTRODUCTION

During the course of developing the CBI numerous test were performed to monitor execution speed, memory utilization, and the accuracy of the CBI's interpretation of each of the automatically parsed captions that are located in Appendix E. Table 1 provides a qualitative summary of the test results. The table has seven columns: the first three contain information provided in Appendix D. The remaining four contain information taken from the built-in statistics functions in C-Prolog and Quintus Prolog.

In order to make some comparison between the captions we have added the word count column and the depth of the parse tree column. The word count column refers to how many words were in the caption to include punctuation marks, i.e., commas and periods. The depth of the parse tree column was provided to show the complexity of the caption. The remaining columns will be discussed below.

B. EXECUTION SPEED

The focus of this thesis was not the speed or efficiency of the parser except as a measure to compare the implemented grammar rules. The times provide a useful tools in which to compare the grammar structure of the various captions. The execution time for the two versions of Prolog taken from each version's statistical functions are provided in seconds of wall clock time. It should be noted that the compiled versions of Prolog outperformed the interpretive Prolog except for a few instances which are discussed in the next section. We did have five captions for which we did not get any results and are annotated in Table 1 with "----" in the time columns. This was due to running out of heap stack space and will be discussed in the next section.

C. MEMORY REQUIREMENTS

The C-Prolog and Quintus Prolog are significantly different in the way that they allocate memory. This is important to this thesis because a major concern and problem area for us, while working with C-Prolog, was that we would run out of heap stack space. C-Prolog uses a static memory allocation scheme to provide memory to its various stacks [Ref. 12:p. 33]. The problem with this is that if a stack runs out of memory space, the program terminates prematurely. After extensive testing we found that the heap stack had to be increased to as much as 3500 kilobytes in order to parse 25 out of the 30 captions. The other five sentences failed to run to completion because the heap stack reached its limit first. Table 1 provides the memory that was required for each caption in the C-Prolog memory column.

Quintus Prolog provides memory allocation dynamically [Ref. 13:p. 121]. Table 1 give the total memory (in kilobytes) that was required for each caption.

We sought to resolve the problem of running out of heap space by using Quintus Prolog to compile our code. While Quintus Prolog gave us some increased performance in time for most captions as noted in the previous section, we were surprised to find that we received some results that were worse than those for C-Prolog. We believe that because Quintus dynamically allocates memory to its global stack (which contains the heap stack and the trail stack) there are a lot of calls to the operating system to request more memory space once the program has increased to a significant size.

D. OUTPUT ACCURACY

The accuracy of the meaning-list output for the captions is dependent on all of the different components of the system as defined in the previous chapter. The accuracy for the system divided between syntactic accuracy and semantic accuracy. Syntactically the system is completely accurate. Each of the grammar rules were tested individually and with their parent rule if they had one. As far as the semantic accuracy is concerned, only about a few of the captions are completely correct. The semantic interpretation work is beyond the scope of this thesis.

TABLE 1 - SUMMARY OF TESTING RESULTS

sent#	word count	Depth of parse tree	C-Prolog exec time	Quintus exec time	C-Prolog memory	Quintus memory
1	7	5	2.13	0.77	217	398
2	7	4	3.08	1.14	229	398
3	9	7	3.45	1.15	244	398
4	9	5	3.52	1.28	254	398
5	10	7	4.00	1.33	259	398
6	15	8	9.57	4.35	309	422
7	12	6	8.08	2.62	282	398
8	15	8	21.08	11.93	340	398
9	18	8	116.18	203.60	880	722
10	15	8	499.02	1080.30	1297	1022
11	16	8	9.80	5.02	342	422
12	17	10	12.97	4.30	330	398
13	18	8	11.17	4.50	340	398
14	16	9	16.17	8.40	397	398
15	22	11	106.80	161.88	810	646
16	23	12	20.78	13.57	461	398
17	26	9	67.77	55.42	590	458
18	23	8	----	13053.50	----	4202
19	19	9	2569.68	8958.00	2771	1990
20	21	11	1021.05	2475.38	1833	1362
21	19	7	65.33	60.20	613	506
22	29	11	574.07	1486.97	1873	1306
23	20	8	47.47	32.18	454	426
24	30	10	----	----	----	----
25	26	11	2446.40	6301.17	3590	2474
26	31	11	----	----	----	----
27	28	13	163.78	174.57	988	418
28	36	11	1148.32	3099.13	2763	1842
29	32	12	----	----	----	----
30	24	8	----	----	----	----

VI. CONCLUSIONS

A. ACHIEVEMENTS

The major goal of this thesis was to test the premise that captions are a subset of the English language and therefore they should have a relatively simple set of grammar rules. Our research found that captions were indeed a subset of the English language but that subset was still fairly large. After thoroughly analyzing each of the captions we found that they varied more than we had anticipated in style and sentence construction.

Even though our premise was overstated we were still able to develop a natural language interface that could handle correctly a large number of the sentences as seen in Chapter V. We were able to prove that the grammar rules that were developed could recognize their intended structures. We were also able to develop the interface far enough in order to be able to continue with the integration of it to the Multimedia Database System.

B. AREAS FOR FURTHER RESEARCH

The test results in the previous chapter revealed that there were some captions that could not be parsed. I believe there are a number of improvements that could be made. The first is that the code that combines the meanings of two parse rules needs to be enhanced in capability. There need to be fewer cases in the default rule. This would help to get a better interpretation of the captions.

The second area for improvement is in the code that checks the semantics of a meaning. At the present time the code only checks the semantics of prepositional phrases. This improvement would also give a better interpretation for the captions and it would possibly improve the parsing speed because it would not allow for constructs that would not be valid.

A major area for further research should be with the dictionary portion of the CBI. At the present time the dictionary is searched sequentially and the first occurrence of the word is returned to be tested for possible acceptance. If a structure later fails another sequential search must be accomplished. This is a very costly procedure in time. Research into changing the dictionary's structure and increasing the information contained could greatly enhance the CBI.

Another major area for research is with the implementation of the code itself. As seen in Chapter V using a compiled version of Prolog significantly improved performance times. I believe that some rewriting of the code to take advantage of a compiled version of Prolog's built-in functions would also improve performance.

LIST OF REFERENCES

1. Thomas, C. A., *A Program Interface Prototype for a Multimedia Database Incorporating Images*, Master's Thesis, Naval Postgraduate School, Monterey, California, December 1988.
2. Sawyer, G. R., *Managing Sound in a Relational Multimedia Database System, Master's Thesis*, Naval Postgraduate School, Monterey, California, December 1988.
3. Naval Postgraduate School Report Number NPS52-89-020, *A Multimedia Database Management System Supporting Contents Search In Meadia Data*, by V. Y. Lum and K. Meyer-Wegener, March 1989.
4. Barr, A. and Feigenbaum, E. A., *The Handbook of Artificial Intelligence*, v.1, HeurisTech Press, 1981.
5. Allen, J., *Natural Language Understanding*, The Benjamin/Cummings Publishing Company, Inc., 1987.
6. Winograd, T., *Language As A Cognitive Process Volume 1: Syntax*, Addison-Wesley Publishing Company, Inc.,1983.
7. Grosz, B. J., and others, "TEAM: An Experiment in the Design of Transportable Natural-Language Interfaces," *Artificial Inlelligence*, v. 32, 1987.
8. Steinberg, R., *World War II: Island Fighting*, Time-Life Books Inc., 1978.
9. Collins, J. L., Jr, *The Marshall Cavendish Illustrated Encyclopedia of World War II*, v. 19, Marshall Cavendish Corporation,1972.
10. Collier, P.F, *The Picture History of World War II 1939-1945*, Grosset and Dunlap Publishers, 1946.
11. Reynolds, C. G., *The Epic of Flight: The Carrier War*, Time-Life Books Inc., 1982.
12. Pereira, F., *C-Prolog User's Manual Version 1.5*, SRI International, February 1990.
13. *Quintus Prolog Reference Manual*, Quintus Computer Systems, Inc., January 1990.

APPENDIX A

THE ORIGINAL CAPTIONS

This appendix contains the original captions that were acquired for the basis for this thesis. All of the captions were taken directly from thier particular reference. The superscript numbers were added the head of each sentence in order to cross reference them with Appendix B.

A. [16]U.S. soldiers wading ashore in columns churn up the waters off Morotai Island, midway between western New Guinea and the Philippines. [14]MacArthur wanted Morotai so Allied aircraft could operate from there and protect his Philippine landings. [2]The Morotai invaders met no resistance. [Ref. 8:p. 150]

B. [26]About to come under attack by planes of the U.S. Navy in November of 1943, Japanese warships maneuver frantically out of the harbor at Rabaul to head for open seas. [Ref. 8:p. 163]

C. [25]Surrounded by bursting bombs, a "Shokaku" class fleet carrier turns sharply in an effort to evade the attentions of U.S. carrier-based strike aircraft. [4]In the foreground destroyers take similar evasive action. [Ref. 9:p. 2587]

D. [13]A "Kongo" class battleship in trouble after being hit by bombs and a fleet carrier turning away. [Ref. 9:p. 2587]

E. [8]An American cruiser, led by a destroyer, maneuvers to avoid Japanese attacks. [Ref. 9:p. 2589]

F. [11]The Japanese heavy cruiser Nachi under air attack in Manila Bay on November 5, 1944. [1]She was sunk in the attack. [Ref. 9:p. 2654]

G. [15]Rabaul Harbor presented this sight after the attack by Allied planes, Nov. 5, 1943, on the Japanese stronghold. [27]Ships burning and sinking litter the harbor, while smoke and flames rise from battered shore installations and warehouses which were bombed during the seventyfive minute engagement. [17]Three enemy destroyers, eight merchant ships, and four coastal vessels were sunk, and sixty-seven Japanese planes shot down in this Allied blow. [Ref. 10:p. 188]

H. [18]Japanese ships scurry to get out of land-locked Rabaul Harbor during the attacks by American bombers on Nov. 5, 1943. [28]Planes from the aircraft carrier Saratoga raided the enemy naval base on New Britain Island and several of the Twenty-five ships there were hit and set afire as they tried to reach the open sea. [19]Rabaul was under constant bombardment by our planes in order to neutralize or destroy that strong Japanese base. [Ref. 10:p. 189]

I. [20]Japanese freighter, smashed by Allied bombers during raid on Rabaul, New Britain, settles fast at the stern. [3]Rabaul was the main Japanese base in the area. [29]From October through December, 1943, Allied air forces inflicted such destruction at Rabaul on ships, planes, and installations that the Japanese abandoned it as a major base. [Ref. 10:p. 189]

J. [9]Japanese carrier makes a frantic fight for life in the Philippine Sea on June 19, 1944. [6]One bomb has just landed near the ship's bow and another at her stern. [7]In the foreground an enemy destroyer gyrates crazily to escape bombs. [5]American planes scored a decisive victory over enemy forces. [Ref. 10:p. 231]

K. [10]Navy Task Force ships land Marines of the Fifth Amphibious Corps on Iwo Jima, Feb. 19, 1945. [21]Bombs and shells from heavy pre-invasion bombardment were still bursting ashore as landing craft hit the beach. [22]LCI(G)'s (Landing Craft Infantry Gunboats) had a cleared way for the Marines by moving into the beaches and firing into shore positions with small caliber guns. [12]But the enemy's big guns, well camouflaged in reinforced caves, were not knocked out. [23]As the Marines stormed ashore, the Japanese opened up, pouring shells from high ground onto our forces. [Ref. 10:p. 262]

L. [24]55,000-ton Missouri, flanked by destroyer, steams into Tokyo Bay, August 28, after waiting two days in Sagami Bay for demining of the upper waters. [Ref. 10:p. 271]

M. [30]Circling wildly, the carrier Soryu is attacked by American dive bombers from Midway as her Zeros try in vain to defend her. [Ref. 11:p. 94]

APPENDIX B

MODIFIED CAPTIONS

This apppendix provids a list of the captions in the order in which they were tested. The superscript letter at the head of each sentences is used for cross referencing to the original caption in Appendix A.

1. [F]She was sunk in the attack.
2. [A]The Morotai invaders met no resistance.
3. [I]Rabaul was the main Japanese base in the area.
4. [C]In the foreground destroyers take similar evasive action.
5. [J]American planes scored a decisive victory over enemy forces.
6. [J]One bomb has just landed near the ships bow and another at her stern.
7. [J]In the foreground an enemy destroyer gyrates crazily to escape bombs.
8. [E]An American cruiser, led by a destroyer, maneuvers to avoid Japanese attacks.
9. [J]Japanese carrier makes a frantic fight for life in the Philippine Sea on June 19, 1944.
10. [K]Navy Task Force ships land Marines of the Fifth Amphibious Corps on Iwo Jima, Feb 19, 1945.
11. [F]The Japanese heavy cruiser Nachi under air attack in Manila Bay on November 5, 1944
12. [K]But the enemy's big guns, well camouflaged in reinforced caves, were not knocked out.
13. [D]A Kongo class battleship in trouble after being hit by bombs and a fleet carrier turning away.

14. [A]MacArthur wanted Morotai so Allied aircraft could operate from there and protect his Philippine landings.

15. [G]Rabaul Harbor presented this sight after the attack by Allied planes, Nov 5, 1943, on the Japanese stronghold.

16. [A]US soldiers wading ashore in columns churn up the waters off Morotai Island, midway between western New Guinea and the Philippines.

17. [G]Three enemy destroyers, eight merchant ships, and four coastal vessels were sunk, and 67 Japanese planes shot down in this Allied blow.

18. [H]Japanese ships scurry to get out of land-locked Rabaul Harbor during the attacks by American bombers on Nov 5, 1943.

19. [H]Rabaul was under constant bombardment by our planes in order to neutralize or destroy that strong Japanese base.

20. [I]Japanese freighter, smashed by Allied bombers during raid on Rabaul, New Britain, settles fast at the stern.

21. [K]Bombs and shells from heavy pre-invasion bombardment were still bursting ashore as landing craft hit the beach.

22 [K]LCI (Landing Craft Infantry Gunboats) had cleared the way for the Marines by moving into the beaches and firing into shore positions with small caliber guns.

23. [K]As the Marines stormed ashore, the Japanese opened up, pouring shells from high ground onto our forces.

24. [L]55000 ton Missouri, flanked by destroyer, steams into Tokyo Bay, August 28, after waiting two days in Sagami Bay for demining of the upper waters.

25. [C]Surrounded by bursting bombs, a Shokaku class fleet carrier turns sharply in an effort to evade the attentions of US carrier-based strike aircraft.

26. [B]About to come under attack by planes of the US Navy in November 1943, Japanese warships maneuver frantically out of the harbor at Rabaul to head for open seas.

27. [G]Ships burning and sinking litter the harbor, while smoke and flames rise from battered shore installations and warehouses which were bombed during the 75 minute engagement.

28. [H]Planes from the aircraft carrier Saratoga raided the enemy naval base on New Britain Island and several of the 25 ships there were hit and set afire as they tried to reach the open sea.

29. [I]From October through December, 1943, Allied air forces inflicted such destruction at Rabaul on ships, planes, and installations that the Japanese abandoned it as a major base.

30. [M]Circling wildly, the carrier Soryu is attacked by American dive bombers from Midway as her Zeros try in vain to defend her.

APPENDIX C

ABBREVIATION LIST

This appendix lists all of the abbreviations that are used throughout this thesis.

adj	adjective
adjl	adjective list
adv	adverb
apb	apositive begining
aps	apostrophe
aux	auxiliary
c_dt_c	comma with date with comma
c_geo	comma with geographical location
c_np	comma with noun phrase
c_pps_c	comma with multiple prepositional phrases with comma
c_prtp	comma with participle phrase
c_prtp_c	comma with participle phrase with comma
c_yr	comma with year
cj	common conjuction
cj_mo	conjuntion with month
cj_np	conjunction with noun phrase
cj_prt	conjunction with participle
cj_snt	conjunction with sentence
cj_vp	conjunction with verb phrase
cls	clause
cls_c	clause with comma
clscj	clause conjuntion
com	comma
cp	close parenthesis
cp_np	complex noun phrase
cp_prt	complex participle
cp_snt	complex sentence
cp_vp	complex verb phrase
d	determiner

day	day
dblcj	double conjuntion
dt	date
dt_c	date with comma
g	gerand
geo	geographic location
i	infinitive
im	infinitive marker
ip	infinitive phrase
ip_c	infinitive pharse with comma
lv	linking verb
mo	month
n	noun
ng	noun group
np	noun phrase
num	number
op	open parenthesis
op_np	open parenthesis with noun phrase
p	preposition
perprn	personal pronoun
posprn	possive pronoun
pn	proper noun
pp	preposition phrase
pps	multiple prepositional phrases
pps_c	multiple prepositional phrases with comma
prn	pronoun
prt	participle
prtp	participle phrase
prtp_c	participle phrase with comma
se	sentence end
snt	sentence
v	verb
vg	verb group
vp	verb phrase
yr	year

APPENDIX D

MANUALLY PARSED CAPTIONS

This appendix provides the manually parsed sentence. The captions are represented in a binary parse tree.

1. ᶠShe was sunk in the attack.

```
snt --> snt, se
    snt --> np, vp
          np --> ng
                ng --> prn
                      prn --> She
          vp --> vg, pp
                vg --> lv, v
                      lv --> was
                      v --> sunk
                pp --> p, np
                      p --> in
                      np --> d , n
                            d --> the
                            n --> attack
    se --> .
```

2. ^AThe Morotai invaders met no resistance.

```
snt --> snt, se
    snt --> np,  vp
          np --> d , ng
                d --> **The**
                ng --> adj,  n
                      adj --> **Morotai**
                      n --> **invaders**
          vp --> v,  ng
                v --> **met**
                ng --> adj,  n
                      adj --> **no**
                      n --> **resistance**
    se --> .
```

3. 'Rabaul was the main Japanese base in the area.

```
snt --> snt, se
    snt --> geo, vp
          geo --> Rabaul
          vp --> v, np
                v --> was
                np --> np, pp
                      np --> d, ng
                            d --> the
                            ng --> adjl, n
                                  adjl --> adj, adj
                                        adj --> main
                                        adj --> Japanese
                                  n --> base
                      pp --> p, np
                            p --> in
                            np --> d, n
                                  d --> the
                                  n --> area
    se --> .
```

4. [c]In the foreground destroyers take similar evasive action.

```
snt --> snt, se
    snt --> np, vp
          np --> pp,  n
                pp --> p,  np
                      p --> In
                      np --> d,  n
                            d --> the
                            n --> foreground
                n --> destroyers
          vp --> v,  ng
                v --> take
                ng --> adjl, n
                      adjl --> adj,  adj
                            adj --> similar
                            adj --> evasive
                      n --> action
    se --> .
```

5. ᴶAmerican planes scored a decisive victory over enemy forces.

```
snt --> snt, se
    snt --> ng, vp
          ng --> adj, n
                adj --> American
                n --> planes
          vp --> v, np
                v --> scored
                np --> np, pp
                      np --> d, ng
                            d --> a
                            ng --> adj, n
                                  adj --> decisive
                                  n --> victory
                      pp --> p, ng
                            p --> over
                            ng --> adj, n
                                  adj --> n
                                        n --> enemy
                                  n --> forces
    se --> .
```

6. ᴶOne bomb has just landed near the ships bow and ano'her at her stern.

```
snt --> snt, se
    snt --> ng, vp
          ng --> adj, n
                adj --> **One**
                n --> **bomb**
          vp --> vg, np
                vg --> aux, vg
                      aux --> **has**
                      vg --> adv, v
                            adv --> **just**
                            v --> **landed**
                np --> np, cj_np
                      np --> adj, np
                            adj --> **near**
                            np --> d, ng
                                  d --> **the**
                                  ng --> adj, n
                                        adj --> **ship's**
                                        n --> **bow**
                      cj_np --> cj, np
                            cj --> **and**
                            np --> prn, pp
                                  prn --> **another**
                                  pp --> p, ng
                                        p --> **at**
                                        ng --> prn, n
                                              prn --> **her**
                                              n --> **stern**
    se --> .
```

7. [J]In the foreground an enemy destroyer gyrates crazily to escape bombs.

```
snt --> snt,  se
    snt --> pp,  snt
          pp --> p,  np
                p --> In
                np --> d,  n
                      d --> the
                      n --> foreground
          snt --> np,  vp
                np --> d,  ng
                      d --> an
                      ng --> adj,  n
                            adj --> n
                                  n --> enemy
                            n --> destroyer
                vp --> vg,  ip
                      vg --> v,  adv
                            v --> gyrates
                            adv --> crazily
                      ip --> i,  n
                            i --> im,  v
                                  im --> to
                                  v --> escape
                            n --> bombs
   se --> .
```

8. [E]An American cruiser, led by a destroyer, maneuvers to avoid Japanese attacks.

```
snt --> snt, se
    snt --> np, vp
          np --> np, c_prtp_c
                np --> d, ng
                      d --> An
                      ng --> adj, n
                            adj --> American
                            n --> cruiser
                c_prtp_c --> com, prtp_c
                      com --> ,
                      prtp_c --> prtp, com
                            prtp --> prt, pp
                                  prt --> led
                                  pp --> p, np
                                        p --> by
                                        np --> d, n
                                              d --> a
                                              n --> destroyer
                            com --> ,
          vp --> v, ip
                v --> maneuvers
                ip --> i, ng
                      i --> im, v
                            im --> to
                            v --> avoid
                      ng --> adj, n
                            adj --> Japanese
                            n --> attacks
    se --> .
```

9. ᴶJapanese carrier makes a frantic fight for life in the Philippine Sea on June 19, 1944.

```
snt --> snt, se
   snt --> ng, vp
        ng --> adj, n
             adj --> Japanese
             n --> carrier
        vp --> vp, pp
             vp --> v, np
                  v --> makes
                  np --> np, pp
                       np --> np, pp
                            np --> d, ng
                                 d --> a
                                 ng --> adj, n
                                      adj --> frantic
                                      n --> fight
                            pp --> p, n
                                 p --> for
                                 n --> life
                       pp --> p, np
                            p --> in
                            np --> d, geo
                                 d --> the
                                 geo --> Philippine Sea
             pp --> p, dt
                  p --> on
                  dt --> dt, c_yr
                       dt --> mo, num
                            mo --> June
                            num --> 19
                       c_yr --> com, num
                            com --> ,
                            num --> 1944
   se --> .
```

10. ᴷNavy Task Force ships land Marines of the Fifth Amphibious Corps on Iwo Jima, Feb 19, 1945.

```
snt --> snt, se
    snt --> snt, c_dt
          snt --> ng, vp
                ng --> adj, n
                      adj --> pn
                            pn --> **Navy Task Force**
                      n --> **ships**
                vp --> vp, pp
                      vp --> v, np
                            v --> **land**
                            np --> pn, pp
                                  pn --> **Marines**
                                  pp --> p, np
                                        p --> **of**
                                        np --> d, pn
                                              d --> **the**
                                              pn --> **Fifth Amphibious Corps**
                      pp --> p, geo
                            p --> **on**
                            geo --> **Iwo Jima**
          c_dt --> com, dt
                com --> ,
                dt --> dt, c_yr
                      dt --> mo, num
                            mo --> **Feb**
                            num --> **19**
                      c_yr --> com, num
                            com --> ,
                            num --> **1945**
    se --> .
```

11. [F]The Japanese heavy cruiser Nachi under air attack in Manila Bay on November 5, 1944

```
npc --> np, se
    np --> np, pp
        np --> np, pps
            np --> d, ng
                d --> The
                ng --> adjl, pn
                    adjl --> adj, adjl
                        adj --> Japanese
                        adjl --> adj, adj
                            adj --> heavy
                            adj --> n
                                n --> cruiser
                    pn --> Nachi
            pps --> pp, pp
                pp --> p, ng
                    p --> under
                    ng --> adj, n
                        adj --> n
                            n --> air
                        n --> attack
                pp --> p, geo
                    p --> in
                    geo --> Manila Bay
        pp --> p, dt
            p --> on
            dt --> dt, c_yr
                dt --> mo, num
                    mo --> November
                    num --> 5
                c_yr --> com, num
                    com --> ,
                    num --> 1944
    se --> .
```

12. ᴷBut the enemy's big guns, well camouflaged in reinforced caves, were not knocked out.

```
snt --> snt, se
    snt --> cj_snt
        cj_snt --> cj, snt
            cj --> But
            snt --> np, vg
                np --> np, c_prtp_c
                    np --> d, ng
                        d --> the
                        ng --> adjl, n
                            adjl --> adj, adj
                                adj --> enemy's
                                adj --> big
                            n --> guns
                    c_prtp_c --> com, prtp_c
                        com --> ,
                        prtp_c --> prtp, com
                            prtp --> prtp, pp
                                prtp --> adv, prt
                                    adv --> well
                                    prt --> camouflaged
                                pp --> p, ng
                                    p --> in
                                    ng --> adj, n
                                        adj --> reinforced
                                        n --> caves
                            com --> ,
                vg --> lv, vg
                    lv --> were
                    vg --> adv, vg
                        adv --> not
                        vg --> v, adv
                            v --> knocked
                            adv --> out
    se --> .
```

13. ^{D}A Kongo class battleship in trouble after being hit by bombs and a fleet carrier turning away.

```
npc --> np, se
    np --> np, cj_np
        np --> np, pp
            np --> np, pp
                np --> d, ng
                    d --> A
                    ng --> adjl, n
                        adjl --> adj, adj
                            adj --> pn
                                pn --> Kongo
                            adj --> n
                                n --> class
                        n --> battleship
                pp --> p, n
                    p --> in
                    n --> trouble
            pp --> p, np
                p --> after
                np --> n, pp
                    n --> g
                        g --> vg
                            vg --> aux, v
                                aux --> being
                                v --> hit
                    pp --> p, n
                        p --> by
                        n --> bombs
        cj_np --> cj, np
            cj --> and
            np --> np, prtp
                np --> d, ng
                    d --> a
                    ng --> adj, n
                        adj --> n
                            n --> fleet
                        n --> carrier
                prtp --> prt, n
                    prt --> turning
                    n --> away
    se --> .
```

14. ᴬMacArthur wanted Morotai so Allied aircraft could operate from there and protect his Philippine landings.

```
snt --> snt, se
    snt --> snt, cj_snt
        snt --> pn, vp
            pn --> MacArthur
            vp --> v, geo
                v --> wanted
                geo --> Morotai
        cj_snt --> cj, snt
            cj --> so
            snt --> ng, vp
                ng --> adj, n
                    adj --> Allied
                    n --> aircraft
                vp --> vp, cj_vp
                    vp --> vg, pp
                        vg --> aux, v
                            aux --> could
                            v --> operate
                        pp --> p, pn
                            p --> from
                            pn --> there
                    cj_vp --> cj, vp
                        cj --> and
                        vp --> v, np
                            v --> protect
                            np --> prn, ng
                                prn --> his
                                ng --> adj, n
                                    adj --> Philippine
                                    n --> landings
    se --> .
```

15. ^{G}Rabaul Harbor presented this sight after the attack by Allied planes, Nov 5, 1943, on the Japanese stronghold.

```
snt --> snt, se
    snt --> geo, vp
          geo --> **Rabaul Harbor**
          vp --> vp, pp
                vp --> v, ng
                      v --> presented
                      ng --> adj, n
                            adj --> this
                            n --> sight
                pp --> p, np
                      p --> after
                      np --> np, pps
                            np --> d, n
                                  d --> the
                                  n --> attack
                            pps --> pp, pp
                                  pp --> pp, c_dt_c
                                        pp --> p, ng
                                              p --> by
                                              ng --> adj, n
                                                    adj --> Allied
                                                    n --> planes
                                        c_dt_c --> c_dt, com
                                              c_dt --> com, dt
                                                    com --> ,
                                                    dt --> dt, c_yr
                                                          dt --> mo, num
                                                                mo --> Nov
                                                                num --> 5
                                                          c_yr --> com, num
                                                                com --> ,
                                                                num --> 1943
                                  pp --> p, np
                                        p --> on
                                        np --> d, ng
                                              d --> the
                                              ng --> adj, n
                                                    adj --> Japanese
                                                    n --> stronghold
   se --> .
```

16. ᴬUS soldiers wading ashore in columns churn up the waters off Morotai Island, midway between western New Guinea and the Philippines.

```
snt --> snt, se
    snt --> np, vp
          np --> ng, prtp
                ng --> adj, n
                      adj --> US
                      n --> soldiers
                prtp --> prtp, pp
                      prtp --> prt, adv
                            prt --> wading
                            adv --> ashore
                      pp --> p, n
                            p --> in
                            n --> columns
          vp --> vg, np
                vg --> v, adv
                      v --> churn
                      adv --> up
                np --> np, pp
                      np --> d, n
                            d --> the
                            n --> waters
                      pp --> p, np
                            p --> off
                            np --> geo, pp
                                  geo --> Mortotai Island
                                  c_pp --> com, pp
                                        com --> ,
                                        pp --> adv, pp
                                              adv --> midway
                                              pp --> p, np
                                                    p --> between
                                                    np --> ng, cj_np
                                                          ng --> adj, geo
                                                                adj --> western
                                                                geo --> New Guinea
                                                          cj_np --> cj, np
                                                                cj --> and
                                                          np --> d, geo
                                                                 d --> the
                                                                 geo --> Philippines
          se --> .
```

17. [G]Three enemy destroyers, eight merchant ships, and four coastal vessels were sunk, and 67 Japanese planes shot down in this Allied blow.

```
snt --> snt, se
    snt --> snt, pp
          snt --> snt, cj_snt
                snt --> ng, vg
                      ng --> ng, cj_ng
                            ng --> adjl, n
                                  adjl --> adj, adj
                                        adj --> Three
                                        adj --> n
                                              n --> enemy
                                  n --> destroyers
                            cj_ng --> cj, ng
                                  cj --> com
                                        com --> ,
                                  ng --> adjl, n
                                        adjl --> adj, adj
                                              adj --> eight
                                              adj --> n
                                                    n --> merchant
                                        n --> ships
                      cj_ng --> cj, ng
                            cj --> com, cj
                                  com --> ,
                                  cj --> and
                            ng --> adjl, n
                                  adjl --> adj, adj
                                        adj --> four
                                        adj --> coastal
                                  n --> vessels
                cj_snt --> cj, snt
                      cj --> com, cj
                            com --> ,
                            cj --> and
                      snt --> ng, vg
                            ng --> adjl, n
                                  adjl --> adj, adj
                                        adj --> num
                                              num --> 67
                                        adj --> Japanese
                                  n --> planes
```

```
                        vg --> v,  adv
                                  v --> **shot**
                                  adv --> **down**
         pp --> p,  ng
                   p --> **in**
                   ng --> adjl,  n
                            adjl --> adj,  adj
                                      adj --> **this**
                                      adj --> **Allied**
                            n --> **blow**
se --> .
```

18. [H]Japanese ships scurry to get out of land-locked Rabaul Harbor during the attacks by American bombers on Nov 5, 1943.

```
snt --> snt,  se
    snt --> ng,  vp
          ng --> adj,  n
                adj --> Japanese
                n --> ships
          vp --> vp,  pp
                vp --> v,  ip
                      v --> scurry
                      ip --> i,  pp
                            i --> im,  vg
                                  im --> to
                                  vg --> v,  adv
                                        v --> get
                                        adv --> out
                            pp --> p,  ng
                                  p --> of
                                  ng --> adj,  geo
                                        adj --> land-locked
                                        geo --> Rabaul Harbor
                pp --> pp,  pp
                      pp --> p,  np
                            p --> during
                            np --> np,  pp
                                  np --> d,  n
                                        d --> the
                                        n --> attacks
                                  pp --> p,  ng
                                        p --> by
                                        ng --> adj,  n
                                              adj --> American
                                              n --> bombers
                      pp --> p,  dt
                            p --> on
                            dt --> dt,  c_yr
                                  dt --> mo,  num
                                        mo --> Nov
                                        num --> 5
```

```
                    c_yr --> com,  num
                         com --> ,
                         num --> 1943
se --> .
```

19. ᴴRabaul was under constant bombardment by our planes in order to neutralize or destroy that strong Japanese base.

```
snt --> snt,  se
    snt --> geo,  vp
        geo --> Rabaul
        vp --> vp,  ip
            vp --> v,  pp
                v --> was
                pp --> p,  np
                    p --> under
                    np --> adj, np
                        adj --> constant
                        np --> n,  pp
                            n --> bombardment
                            pp --> p,  ng
                                p --> by
                                ng --> adj, n
                                    adj --> our
                                    n --> planes
            ip --> ip,  ng
                ip --> pp,  i
                    pp --> p, n
                        p --> in
                        n --> order
                    i --> im, v
                        im --> to
                        v --> v,  cj_v
                            v --> neutralize
                            cj_v --> cj,  v
                                cj --> or
                                v --> destroy
                ng --> adj,  ng
                    adj --> prn
                        prn --> that
                    ng --> adjl,  n
                        adjl --> adj,  adj
                            adj --> strong
                            adj --> Japanese
                        n --> base
    se --> .
```

20. 'Japanese freighter, smashed by Allied bombers during raid on Rabaul, New Britain, settles fast at the stern.

```
snt --> snt,  se
    snt --> np,  vp
          np --> ng,  c_prtp_c
                ng --> adj,  n
                      adj --> Japanese
                      n --> freighter
                c_prtp_c --> com,  prtp_c
                      com --> ,
                      prtp_c --> prtp,  com
                            prtp --> prtp,  pp
                                  prtp --> prt,  pp
                                        prt --> smashed
                                        pp --> p,  ng
                                              p --> by
                                              ng --> adj,  n
                                                    adj --> Allied
                                                    n --> bombers
                                  pp --> p,  np
                                        p --> during
                                        np --> n,  pp
                                              n --> raid
                                              pp --> p, geo
                                                    p --> on
                                                    geo --> geo,  c_geo
                                                          geo --> Rabaul
                                                          c_geo --> com, geo
                                                                com --> ,
                                                                geo --> New Britain
                            com --> ,
          vp --> vg,  pp
                vg --> v,  adv
                      v --> settles
                      adv --> fast
                pp --> p,  np
                      p --> at
                      np --> d,  n
                            d --> the
                            n --> stern
   se --> .
```

21. [K]Bombs and shells from heavy pre-invasion bombardment were still bursting ashore as landing craft hit the beach.

```
snt --> snt, se
    snt --> snt, cls
          snt --> np, vg
                np --> n, pp
                      n --> n, cj_n
                            n --> Bombs
                            cj_n --> cj, n
                                  cj - > and
                                  n --> shells
                      pp --> p, ng
                            p --> from
                            ng --> adjl, n
                                  adjl --> adj, adj
                                        adj --> heavy
                                        adj --> pre-invasion
                                  n --> bombardment
                vg --> vg, adv
                      vg --> aux, vg
                            aux --> were
                            vg --> adv, v
                                  adv --> still
                                  v --> bursting
                      adv --> ashore
          cls --> cj, snt
                cj --> as
                snt --> ng, vp
                      ng --> adj, n
                            adj --> landing
                            n --> craft
                      vp --> v, np
                            v --> hit
                            np --> d, n
                                  d --> the
                                  n --> beach
    se --> .
```

22. ᴷLCI (Landing Craft Infantry Gunboats) had cleared the way for the Marines by moving into the beaches and firing into shore positions with small caliber guns.

```
snt --> snt, se
    snt --> pn, vp
        pn --> pn, ape
            pn --> LCI
            ape --> apb, cp
                apb --> op, pn
                    op --> (
                    pn --> Landing Craft Infantry Gunboats
                cp --> )
        vp --> vp, pp
            vp --> vg, np
                vg --> aux, v
                    aux --> had
                    v --> cleared
                np --> np, pp
                    np --> d, n
                        d --> the
                        n --> way
                    pp --> p, np
                        p --> for
                        np --> d, pn
                            d --> the
                            pn --> Marines
            pp --> p, np
                p --> by
                np --> np, cj_np
                    np --> n, pp
                        n --> g
                            g --> moving
                        pp --> p, np
                            p --> into
                            np --> d, n
                                d --> the
                                n --> beaches
                    cj_np --> cj, np
                        cj --> and
                        np --> np, pp
                            np --> n, pp
                                n --> g
                        g --> **firing**
                    pp --> p,  ng
                        p --> **into**
                        ng --> adj,  n
                            adj --> n
                                n --> **shore**
                            n --> **positions**
                pp --> p,  ng
                    p --> **with**
                    ng --> adjl,  n
                        adjl --> adj,  adj
                            adj --> **small**
                            adj --> **caliber**
                        n --> **guns**
se --> .
```

23. KAs the Marines stormed ashore, the Japanese opened up, pouring shells from high ground onto our forces.

```
snt --> snt, se
    snt --> cls_c, snt
        cls_c --> cls, com
            cls --> cj, snt
                cj --> As
                snt --> np, vg
                    np --> d, pn
                        d --> the
                        pn --> Marines
                    vg --> v, adv
                        v --> stormed
                        adv --> ashore
            com --> ,
        snt --> snt, c_prtp_c
            snt --> np, vp
                np --> d, pn
                    d --> the
                    pn --> Japanese
                vp --> v, adv
                    v --> opened
                    adv --> up
            c_prtp_c --> com, prtp
                com --> ,
                prtp --> prtp, pp
                    prtp --> prtp, pp
                        prtp --> prt, n
                            prt --> pouring
                            n --> shells
                        pp --> p, ng
                            p --> from
                            ng --> adj, n
                                adj --> high
                                n --> ground
                    pp --> p, ng
                        p --> onto
                        ng --> adj, n
                            adj --> our
                            n --> forces
    se --> .
```

24. ᴸ55000 ton Missouri, flanked by destroyer, steams into Tokyo Bay, August 28, after waiting two days in Sagami Bay for demining of the upper waters.

```
snt --> snt, se
    snt --> np, vp
        np --> ng, c_prtp_c
            ng --> adjl, pn
                adjl --> adj, adj
                    adj --> num
                        num --> 55000
                    adj --> n
                        n --> ton
                pn --> Missouri
            c_prtp_c --> com, prtp_c
                com --> ,
                prtp_c --> prtp, com
                    prtp --> prt, pp
                        prt --> flanked
                        pp --> p, n
                            p --> by
                            n --> destroyer
                    com --> ,
        vp --> vp, pp
            vp --> vp, c_dt_c
                vp --> v, pp
                    v --> steams
                    pp --> p, geo
                        p --> into
                        geo --> Tokyo Bay
                c_dt_c --> c_dt, com
                    c_dt --> com, dt
                        com --> ,
                        dt --> mo, num
                            mo --> August
                            num --> 28
                    com --> ,
            pp --> p, np
                p --> after
                np --> np, pp
                    np --> ng, pp
                        ng --> n, ng
                            n --> g
                                g --> waiting
                    ng --> adj, n
                              adj --> two
                              n --> days
          pp --> p, geo
                    p --> in
                    geo --> Sagami Bay
pp --> p, np
          p --> for
          np --> n, pp
                    n --> g
                              g --> demining
                    pp --> p, np
                              p --> of
                              np --> d, ng
                                        d --> the
                                        ng --> adj, n
                                                  adj --> upper
                                                  n --> waters
se --> .
```

25. [c]Surrounded by bursting bombs, a Shokaku class fleet carrier turns sharply in an effort to evade the attentions of US carrier-based strike aircraft.

```
snt --> snt,  se
    snt --> np,  vp
          np --> prtp_c,  np
                prtp_c --> prtp,  com
                      prtp --> prt,  pp
                            prt --> Surrounded
                            pp --> p,  ng
                                  p --> by
                                  ng --> adj,  n
                                        adj --> bursting
                                        n --> bombs
                      com --> ,
                np --> d,  ng
                      d --> a
                      ng --> adjl,  n
                            adjl --> adjl,  adj
                                  adjl --> adj,  adj
                                        adj --> pn
                                              pn --> Shokaku
                                        adj --> class
                                  adj --> fleet
                            n --> carrier
          vp --> vg,  pp
                vg --> v,  adv
                      v --> turns
                      adj --> sharply
                pp --> p,  np
                      p --> in
                      np --> np,  ip
                            np --> d,  n
                                  d --> an
                                  n --> effort
                            ip --> i,  np
                                  i --> im,  v
                                        im --> to
                                        v --> evade
                                  np --> np,  pp
                                        np --> d,  n
                                              d --> the
                                              n --> attentions
```

```
                  pp --> p,  ng
                        p --> **of**
                        ng --> adjl,  n
                              adjl --> adjl,  adj
                                    adjl --> adj,  adj
                                          adj --> US
                                          adj --> carrier-based
                                    adj --> strike
                              n --> aircraft
se --> .
```

26. [B]About to come under attack by planes of the US Navy in November 1943, Japanese warships maneuver frantically out of the harbor at Rabaul to head for open seas.

```
snt --> snt, se
   snt --> np, vp
      np --> ipc, ng
         ipc --> ip, com
            ip --> ip, pp
               ip --> adv, ip
                  adv --> About
                  ip --> i, pp
                     i --> im, v
                        im --> to
                        v --> come
                     pp --> p, n
                        p --> under
                        n --> attack
               pp --> pp, pp
                  pp --> p, np
                     p --> by
                     np --> n, pp
                        n --> planes
                        pp --> p, np
                           p --> of
                           np --> d, ng
                              d --> the
                              ng --> adj, pn
                                 adj --> US
                                 pn --> Navy
                  pp --> p, dt
                     p --> in
                     dt --> mo, pp
                        mo --> November
                        pp --> p, num
                           p --> of
                           num --> 1943
            com --> ,
         ng --> adj, n
            adj --> Japanese
            n --> warships
      vp --> vp, ip
         vp --> vg, pp
```

```
            vg --> v,  adv
                  v --> **maneuver**
                  adv --> **frantically**
            pp --> adv, pp
                  adv --> **out**
                  pp --> p,  np
                        p --> **of**
                        np --> np,  pp
                              np --> d,  n
                                    d --> **the**
                                    n --> **harbor**
                              pp --> p,  geo
                                    p --> **at**
                                    geo --> **Rabaul**
      ip --> i,  pp
            i --> im,  v
                  im --> **to**
                  v --> **head**
            pp --> p,  ng
                  p --> **for**
                  ng --> adj,  n
                        adj --> **open**
                        n --> **seas**
se --> .
```

27. ᴳShips burning and sinking litter the harbor, while smoke and flames rise from battered shore installations and warehouses which were bombed during the 75 minute engagement.

```
snt --> snt, se
    snt --> snt, cj_snt
        snt --> np, vp
            np --> n, prt
                n --> **Ships**
                prt--> prt, cj_prt
                    prt --> **burning**
                    cj_prt --> cj, prt
                        cj --> **and**
                        prt --> **sinking**
            vp --> v, np
                v --> **litter**
                np --> d, n
                    d --> **the**
                    n --> **harbor**
        cj_snt --> cj, snt
            cj --> com, cj
                com --> **,**
                cj --> **while**
            snt --> n, vp
                n --> n, cj_n
                    n --> **smoke**
                    cj_n --> cj, n
                        cj --> **and**
                        n --> **flames**
                vp --> v, pp
                    v --> **rise**
                    pp --> p, np
                        p --> **from**
                        np --> ng, cls
                            ng --> adj, ng
                                adj --> **battered**
                                ng --> ng, cj_n
                                    ng --> adj, n
                                        adj --> n
                                            n --> **shore**
                                        n --> **installations**
                                    cj_n --> cj, n
                                        cj --> **and**
                                n --> **warehouses**
                      cls --> cj, vp
                           cj --> which
                           vp --> vg, pp
                                vg --> aux, v
                                     aux --> were
                                     v --> bombed
                                pp --> p, np
                                     p --> during
                                     np --> d , ng
                                          d --> the
                                          ng --> adj, n
                                               adj --> adj, n
                                                    adj --> 75
                                                    n --> minute
                                               n --> engagement
se --> .
```

28. ^{H}Planes from the aircraft carrier Saratoga raided the enemy naval base on New Britain Island and several of the 25 ships there were hit and set afire as they tried to reach the open sea.

```
snt --> snt, se
    snt --> snt, cj_snt
        snt --> np, vp
            np --> n, pp
                n --> Planes
                pp --> p, np
                    p --> from
                    np --> d, ng
                        d --> the
                        ng --> adjl, pn
                            adjl --> adj, adj
                                adj --> n
                                    n --> aircraft
                                adj --> n
                                    n --> carrier
                            pn --> Saratoga
            vp --> v, np
                v --> raided
                np --> np, pp
                    np --> d, ng
                        d --> the
                        ng --> adjl, n
                            adjl --> adj, adj
                                adj --> n
                                    n --> enemy
                                adj --> naval
                            n --> base
                    pp --> p, geo
                        p --> on
                        geo --> New Britain Island
        cj_snt --> cj, snt
            cj --> and
            snt --> snt, cls
                snt --> np, vg
                    np --> np, adj
                        np --> prn, pp
                            prn --> several
                            pp --> p, np
                                p --> of
```

```
                                        np --> d,  ng
                                              d --> the
                                              ng --> adj,  n
                                                    adj --> num
                                                          num --> 25
                                                    n --> ships
                            adj --> there
                      vg --> aux,  v
                            aux --> were
                            v --> v,  cj_v
                                  v --> hit
                                  cj_v --> cj,  vg
                                        cj --> and
                                        vg --> v,  adv
                                              v --> set
                                              adv --> afire
                cls --> cj,  snt
                      cj --> as
                      snt --> prn,  vp
                            prn --> they
                            vp --> v,  ip
                                  v --> tried
                                  ip --> i,  np
                                        i --> im,  v
                                              im --> to
                                              v --> reach
                                        np --> d,  ng
                                              d --> the
                                              ng --> adj,  n
                                                    adj --> open
                                                    n --> sea
se --> .
```

29. 'From October through December, 1943, Allied air forces inflicted such destruction at Rabaul on ships, planes, and installations that the Japanese abandoned it as a major base.

```
snt --> snt, se
    snt --> pp_c, snt
          pp_c --> pp, com
                pp --> p, dt
                      p --> From
                      dt --> mo, c_yr
                            mo --> m, cj_mo
                                  m --> October
                                  cj_mo --> cj, mo
                                        cj --> through
                                        mo --> December
                            c_yr --> com, yr
                                  com --> ,
                                  yr --> num
                                        num --> 1943
                com --> ,
          snt --> snt, cj_snt
                snt --> ng, vp
                      ng --> adj, ng
                            adj --> Allied
                            ng --> adj, n
                                  adj --> n
                                        n --> air
                                  n --> forces
                      vp --> v, np
                            v --> inflicted
                            np --> np, pp
                                  np --> ng, pp
                                        ng --> adj, n
                                              adj --> such
                                              n --> destruction
                                        pp --> p, geo
                                              p --> at
                                              geo --> Rabaul
                                  pp --> p, n
                                        p --> on
                                        n --> n, cj_n
                                              n --> ships
                                              cj_n --> cj, n
                                        cj --> com
                                              com --> ,
                                        n --> n,  cj_n
                                              n --> planes
                                              cj_n --> cj,  n
                                                    cj --> com,  cj
                                                          com --> ,
                                                          cj --> and
                                                    n --> installations
                  cj_snt --> cj,  snt
                        cj --> that
                        snt --> np,  vp
                              np --> d,  pn
                                    d --> the
                                    pn --> Japanese
                              vp --> v,  np
                                    v --> abandoned
                                    np --> prn,  pp
                                          prn --> it
                                          pp --> p,  np
                                                p --> as
                                                np --> d,  ng
                                                      d --> a
                                                      ng --> adj,  n
                                                            adj --> major
                                                            n --> base
se --> .
```

30. ^{M}Circling wildly, the carrier Soryu is attacked by American dive bombers from Midway as her Zeros try in vain to defend her.

```
snt --> snt, se
  snt --> snt, cls
      snt --> np, vp
          np --> prtp_c, np
              prtp_c --> prtp, com
                  prtp --> prt, adv
                      prt --> Circling
                      adv --> wildly
                  com --> ,
              np --> d, ng
                  d --> the
                  ng --> adj, pn
                      adj --> n
                          n --> carrier
                      pn --> Soryu
          vp --> vg, pp
              vg --> aux, v
                  aux --> is
                  v --> attacked
              pp --> p, np
                  p --> by
                  np --> ng, pp
                      ng --> adj, ng
                          adj --> American
                          ng --> adj, n
                              adj --> dive
                              n --> bombers
                      pp --> p, geo
                          p --> from
                          geo --> Midway
      cls --> cj, snt
          cj --> as
          snt --> ng, vp
              ng --> adj, pn
                  adj --> prn
                      prn --> her
                  pn --> Zeros
              vp --> vp, ip
                  vp --> v, pp
                      v --> try
                  pp --> p,  n
                        p --> **in**
                        n --> **vain**
            ip --> i,  prn
                  i --> im,  v
                        im --> **to**
                        v --> **defend**
                  prn --> **her**
se --> **.**
```

APPENDIX E

GRAMMAR RULES

This appendix contains a detailed listing the current grammar rules that were used to automatically parse the captions. Refer to Appendix A for the abbreviations.

```
caption --> np

snt --> snt, period
snt --> snt, cj_snt
snt --> cj_snt
snt --> cls_c, snt
snt --> snt, cls
snt --> prtp_c, snt
snt --> snt, c_prtp
snt --> snt, aps
snt --> snt, c_dt
snt --> pps_c, snt
snt --> pps, snt
snt --> np, vp

cj_snt --> conjunction, snt
cj_snt --> comma, snt
cj_snt --> comma_cj, snt

comma_cj --> comma, conjunction

np --> determiner, ng
np --> ng
np --> np, cj_np
np --> np, ip
np --> ip_c, ng
np --> np, pps
np --> np, prtp
np --> prtp_c, np
np --> np, aps

cj_np --> comma, ng
cj_np --> comma_cj, np
cj_np --> conjunction, np

ng --> noun
ng --> pronoun
ng --> geo
ng --> dt
ng --> propnoun
ng --> adjl, ng

adjl --> adj
adjl --> adj, adjl
adjl --> adjl, adj ).
adjl --> adj, cj_adjl

cj_adjl --> conjunction, adj

adj --> adjective
adj --> numeric
adj --> ng

vp --> cp_vp
vp --> vp, ip
vp --> vg, pps
vp --> vg, np
vp --> tobe, np
vp --> vg

cp_vp --> vp, cj_vp
cj_vp --> conjunction, vp
```

```
vg --> adverb, vg
vg --> tobe, vg
vg --> aux, vg
vg --> verb
vg --> tobe, participle
vg --> tobe, pps
vg --> verb, adverb

pps --> pp, pp
pps --> adverb, pp
pps --> comma, pps_c
pps --> comma, pps
pps --> pp --> preposition

pps_c --> pps, comma

pp --> preposition, np
pp --> preposition, numeric

prtp --> comma, prtp_c
prtp --> prtp_c
prtp --> b_prtp, pps
prtp --> b_prtp, cj_b_prtp
prtp --> b_prtp

prtp_c --> prtp, comma
c_prtp --> comma, b_prtp
cj_b_prtp --> conjunction, b_prtp

b_prtp --> adverb, participle
b_prtp --> participle, adverb
b_prtp --> participle

ip --> adverb, ip
ip --> ip, np
ip --> ip, pps
ip --> pps, ip
ip_c --> ip, comma
ip --> infinmarker, vg

dt --> dt2, c_yr
dt --> dt2

dt2 --> month, numeric
dt2 --> month
c_yr --> comma, numeric
c_dt --> comma, dt

c_geo --> comma, geo
geo --> propnoun
geo --> propnoun, c_geo

cls --> clausehead, snt
cls --> clausehead, vp
cls_c --> cls, comma

aps --> openpar, apb
aps --> c_np --> comma
aps --> c_dt --> comma
aps --> c_np, comma
aps --> c_dt, comma
apb --> np, closepar
c_np --> comma, np
snt --> doesword, snt
```

APPENDIX F

AUTOMATICALLY PARSED CAPTIONS

This appendix provides the automatically parsed captions from the Capion-Based Interface. The captions are represented with their meaning lists.

1. [F]She was sunk in the attack.

My interpretation is:

perprn(h2,she)
tense(h2,past)
subject(sink,c2)
action(sink,d2)
transitive(d2)
inside(h2,g2)
definite(g2)
attack(g2)

2. [A]The Morotai invaders met no resistance.

My interpretation is:

definite(i2)
name(i2,Morotai)
place(i2)
plural(i2)
invader(i2)
meet(i2,l2)
subject(meet,i2)
action(meet,j2)
tense(j2,past)
cardinality(l2,0)
resistance(l2)

3. [i]Rabaul was the main Japanese base in the area.

My interpretation is:

```
name(q2,Rabaul)
place(q2)
tense(q2,past)
definite(q2)
main(q2)
nationality(q2,japan)
base(q2)
inside(q2,p2)
definite(p2)
region(p2)
```

4. [c]In the foreground destroyers take similar evasive action.

My interpretation is:

```
inside(f2,c2)
definite(c2)
foreground(c2)
plural(f2)
superclasses(e2,destroyer,[warship])
size(f2,-)
take(f2,j2)
subject(take,f2)
action(take,g2)
plural(g2)
tense(g2,present)
similar(j2)
action(j2,movement)
action(j2)
```

5. ^{J}American planes scored a decisive victory over enemy forces.

My interpretation is:
nationality(e2,us)
plural(e2)
plane(e2)
score(e2,n2)
subject(score,e2)
action(score,f2)
tense(f2,past)
indefinite(n2)
action(n2,thought)
victory(n2)
above(n2,m2)
enemy(m2)
plural(m2)
force(m2)

6. ^{J}One bomb has just landed near the ships bow and another at her stern.

My interpretation is:
cardinality(f3,1)
bomb(f3)
tense(f3,past)
singular(f3)
time(e2,0)
subject(land,e2)
action(land,f2)
transitive(f2)
location(f3,j2)
definite(j2)
plural(j2)
ship(j2)
vehicle(j2)
bow(j2)
and([j2,l2],y2)
pronoun(l2,another)
location(y2,x2)
posprn(x2,her)
stern(x2)

7. ^{J}In the foreground an enemy destroyer gyrates crazily to escape bombs.

My interpretation is:

in_period(y2,c2)
definite(c2)
foreground(c2)
indefinite(y2)
enemy(y2)
superclasses(k2,destroyer,[warship])
size(y2,-)
subject(gyrate,y2)
action(gyrate,o2)
singular(o2)
tense(o2,present)
intransitive(o2)
motion(y2,eratic)
infinmarker(y2)
subject(escape,y2)
action(escape,x2)
plural(x2)
tense(x2,present)
transitive(x2)
plural(y2)
bomb(y2)

8. ^{E}An American cruiser, led by a destroyer, maneuvers to avoid Japanese attacks.

My interpretation is:

indefinite(m2)
nationality(m2,us)
superclasses(c2,cruiser,[warship])
size(m2,-)
lead(m2)
action(lead,e2)
tense(e2,past)
transitive(e2)
location(m2,h2)
indefinite(h2)
superclasses(h2,destroyer,[warship])
size(h2,-)
plural(m2)
maneuver(m2)
infinmarker(m2)
subject(avoid,m2)
action(avoid,v2)
plural(v2)
tense(v2,present)
transitive(v2)
name(m2,Japanese)
subject(attack,m2)
action(attack,d2)
singular(d2)
tense(d2,present)
transitive(d2)

9. [J]Japanese carrier makes a frantic fight for life in the Philippine Sea on June 19, 1944.

My interpretation is:
name(f2,Japanese)
carrier(f2)
vehicle(f2)
make(f2,l3)
subject(make,f2)
action(make,g2)
singular(g2)
tense(g2,present)
indefinite(l3)
action(l3,speed(+))
fight(l3)
beneficiary(l3,q2)
life(q2)
inside(q2,b3)
definite(b3)
name(b3,Philippine Sea)
place(b3)
time_spec(q2,s2)
name(s2,June)
month_name(s2)
cardinality(s2,19)
amount(s2)
cardinality(s2,1944)

10. [K]Navy Task Force ships land Marines of the Fifth Amphibious Corps on Iwo Jima, Feb 19, 1945.

My interpretation is:
name(v2,Navy)
name(v2,Task Force)
plural(v2)
ship(v2)
vehicle(v2)
land(v2,l3)
subject(land,v2)
action(land,m7)
plural(m7)
tense(m7,present)
name(l3,Marines)
military(l3)
coagent(l3,p2)
definite(p2)
name(p2,Fifth Amphibious Corps)
location(l3,s2)
name(s2,Iwo Jima)
place(s2)
name(v2,February)
month_name(v2)
cardinality(v2,19)
amount(v2)
cardinality(v2,1945)

11. ᶠThe Japanese heavy cruiser Nachi under air attack in Manila Bay on November 5, 1944

My interpretation is:
definite(j2)
nationality(j2,japan)
weight(j2,+)
superclasses(g2,cruiser,[warship])
size(h2,-)
name(h2,Nachi)
below(j2,l3)
air(l3)
attack(l3)
inside(l3,m2)
name(m2,Manila Bay)
place(m2)
time_spec(j2,q2)
name(q2,November)
month_name(q2)
cardinality(q2,5)
amount(q2)
cardinality(q2,1944)

12. ᴷBut the enemy's big guns, well camouflaged in reinforced caves, were not knocked out.

My interpretation is:
conjunction(z2,and)
definite(z2)
owned_by(z2,c2)
enemy(c2)
size(z2,+)
plural(z2)
gun(z2)
state(z2,+)
camouflage(z2)
action(camouflage,j2)
tense(j2,past)
transitive(j2)
in_period(z2,m2)
action(m2,+)
plural(m2)
geofeature(m2)
tense(z2,subjunctive)
singular(z2)
state(y2,-)
subject(knock,y2)
action(knock,x2)
transitive(x2)
direction(y2,0)

13. [D]A Kongo class battleship in trouble after being hit by bombs and a fleet carrier turning away.

My interpretation is:

indefinite(s2)
name(s2,Kongo)
class(s2)
superclasses(f2,battleship,[warship])
battleship(s2)
size(s2,+)
inside(s2,l2)
trouble(l2)
time(s2,past)
tense(s2,past)
singular(s2)
subject(hit,v2)
action(hit,w2)
transitive(w2)
coagent(s2,y2)
plural(y2)
bomb(y2)
and([y2,f2],t2)
indefinite(f2)
set(f2)
carrier(f2)
vehicle(f2)
turn(f2)
action(turn,j2)
tense(j2,present)
transitive(j2)
direction(f2,+)

14. [A]MacArthur wanted Morotai so Allied aircraft could operate from there and protect his Philippine landings.

My interpretation is:

name(f2,MacAthur)
want(f2,j2)
subject(want,f2)
action(want,g2)
tense(g2,past)
name(j2,Morotai)
place(j2)
and([f2,h2],r2)
nationality(h2,allied)
aircraft(h2)
tense(h2,past)
singular(h2)
subject(operate,t2)
action(operate,u2)
plural(u2)
transitive(u2)
time_src(h2,w2)
location(w2)
and([h2,e2],m2)
protect(e2,n2)
subject(protect,e2)
action(protect,k2)
plural(k2)
tense(k2,present)
posprn(n2,his)
nationality(n2,philippine)
plural(n2)
landing(n2)

15. [G]Rabaul Harbor presented this sight after the attack by Allied planes, Nov 5, 1943, on the Japanese stronghold.

My interpretation is:

name(p2,Rabaul Harbor)
place(p2)
present(p2,t3)
subject(present,p2)
action(present,j2)
tense(j2,past)
this(t3)
sight(t3)
time_spec(t3,q3)
definite(q3)
attack(q3)
location(q3,u2)
nationality(u2,allied)
plural(u2)
plane(u2)
name(p2,November)
month_name(p2)
cardinality(p2,5)
amount(p2)
cardinality(p2,1943)
time_spec(p2,v2)
definite(v2)
nationality(v2,japan)
stronghold(v2)

16. [A]US soldiers wading ashore in columns churn up the waters off Morotai Island, midway between western New Guinea and the Philippines.

My interpretation is:

nationality(r2,us)
plural(r2)
soldier(r2)
wade(r2)
action(wade,d2)
tense(d2,present)
transitive(d2)
place(r2,beach)
inside(r2,g2)
plural(g2)
column(g2)
churn(r2,w3)
subject(churn,r2)
action(churn,q2)
plural(q2)
tense(q2,present)
vert_direction(r2,+)
definite(w3)
plural(w3)
water(w3)
location(w3,e3)
name(e3,Morotai Island)
place(e3)
place(e3,center)
location(e3,w2)
direction(w2,270)
name(w2,New Guinea)
place(w2)
and([w3,z2],e4)
definite(z2)
name(z2,Philippines)
place(z2)

17.[G]Three enemy destroyers, eight merchant ships, and four coastal vessels were sunk, and 67 Japanese planes shot down in this Allied blow.

18. [H]Japanese ships scurry to get out of land-locked Rabaul Harbor during the attacks by American bombers on Nov 5, 1943.

No results due to failure to parse.

My interpretation is:

```
cardinality(f2,3)
enemy(f2)
superclasses(c2,destroyer,[warship])
size(f2,-)
cardinality(f2,8)
merchant(f2)
plural(f2)
ship(f2)
vehicle(f2)
and([f2,j2],n2)
cardinality(j2,4)
terrain(j2,coast)
plural(j2)
ship(j2)
vehicle(j2)
tense(n2,past)
plural(n2)
subject(sink,o2)
action(sink,p2)
transitive(p2)
and([f2,d3],l3)
amount(d3)
cardinality(d3,67)
nationality(d3,japan)
plural(d3)
plane(d3)
subject(shoot,d3)
action(shoot,o3)
tense(o3,past)
transitive(o3)
vert_direction(d3,-)
inside(d3,c3)
this(c3)
nationality(c3,allied)
blow(c3)
```

19.[H]Rabaul was under constant bombardment by our planes in order to neutralize or destroy that strong Japanese base.

My interpretation is:
name(e5,Rabaul)
place(e5)
tense(e5,past)
below(e5,v2)
a-tion(v2,speed(0))
bombardment(v2)
coagent(v2,k2)
posprn(k2,our)
plural(k2)
plane(k2)
inside(e5,q2)
order(q2)
infinmarker(q2)
subject(neutralize,q2)
action(neutralize,r2)
plural(r2)
tense(r2,present)
transitive(r2)
or([u2,c4],c5)
subject(destroy,c4)
action(destroy,d4)
plural(d4)
tense(d4,present)
transitive(d4)
that(e5,g4)
strength(g4,+)
name(g4,Japanese)
subject(base,g4)
action(base,a5)
plural(a5)
tense(a5,present)
transitive(a5)

20. [I]Japanese freighter, smashed by Allied bombers during raid on Rabaul, New Britain, settles fast at the stern.

My interpretation is:
nationality(z7,japan)
ship(z7)
freighter(z7)
smash(z7)
action(smash,g2)
tense(g2,past)
transitive(g2)
coagent(z7,n2)
nationality(n2,allied)
plural(n2)
bomber(n2)
time_spec(n2,l2)
action(l2)
location(n2,o2)
name(o2,Rabaul)
name(o2,New Britain)
place(o2)
subject(settle,z7)
action(settle,r7)
singular(r7)
tense(r7,present)
transitive(r7)
motion(z7,+)
location(z7,y7)
definite(y7)
stern(y7)

21. ᴷBombs and shells from heavy pre-invasion bombardment were still bursting ashore as landing craft hit the beach.

My interpretation is:

plural(a2)
bomb(a2)
and([a2,c2],x2)
plural(c2)
shell(c2)
time_src(x2,m2)
weight(m2,+)
subtype(m2,k2)
prefix(k2,+)
before(k2)
invation(k2)
bombardment(m2)
tense(x2,past)
plural(x2)
frequency(a2,0)
subject(bursting,a2)
action(bursting,l2)
plural(l2)
transitive(l2)
place(a2,beach)
as([a2,l3],z3)
landing(l3)
craft(l3)
hit(l3,j3)
subject(hit,l3)
action(hit,m3)
plural(m3)
tense(m3,present)
definite(j3)
geofeature(j3)

22. ᴷLCI (Landing Craft Infantry Gunboats) had cleared the way for the Marines by moving into the beaches and firing into shore positions with small caliber guns.

My interpretation is:

name(h2,LCI(G))
clear(h2,w4)
tense(g2,past)
singular(g2)
subject(clear,h2)
action(clear,i2)
definite(w4)
way(w4)
beneficiary(w4,a7)
definite(a7)
name(a7,Marines)
military(a7)
coagent(a7,r2)
move(r2)
inside(w4,u2)
definite(u2)
geofeature(u2)
and([w4,o2],b19)
fire(o2)
inside(o2,a2)
shore(a2)
plural(a2)
position(a2)
coagent(a2,x2)
size(x2,-)
caliber(x2)
plural(x2)
gun(x2)

23. [K]As the Marines stormed ashore, the Japanese opened up, pouring shells from high ground onto our forces.

My interpretation is:

```
coagent(b4,v2)
name(v2,Marines)
military(v2)
storm(v2)
action(storm,h2)
tense(h2,past)
transitive(h2)
place(v2,beach)
definite(v2)
name(v2,Japanese)
open(v2)
action(open,n2)
tense(n2,past)
transitive(n2)
vert_direction(v2,+)
pour(v2)
action(pour,w2)
tense(w2,present)
transitive(w2)
plural(b4)
shell(b4)
source(b4,x3)
height(x3,+)
ground(x3)
above(x3,w3)
posprn(w3,our)
subject(force,b4)
action(force,c4)
singular(c4)
tense(c4,present)
transitive(c4)
```

24. [L]55000 ton Missouri, flanked by destroyer, steams into Tokyo Bay, August 28, after waiting two days in Sagami Bay for demining of the upper waters.

No results.

25. ᶜSurrounded by bursting bombs, a Shokaku class fleet carrier turns sharply in an effort to evade the attentions of US carrier-based strike aircraft.

My interpretation is:

```
surround(r3)
action(surround,b2)
tense(b2,past)
transitive(b2)
location(r3,g2)
action(g2,+)
plural(g2)
bomb(g2)
indefinite(r3)
name(r3,Shokaku)
class(r3)
set(r3)
carrier(r3)
vehicle(r3)
plural(r3)
turn(r3)
sharp(r3,0)
inside(r3,o2)
indefinite(o2)
effort(o2)
infinmarker(o2)
subject(evade,o2)
action(evade,a2)
plural(a2)
tense(a2,present)
transitive(a2)
definite(o2)
plural(o2)
attention(o2)
coagent(r3,u2)
nationality(u2,us)
name(u2,carrier-based)
coagent(u2)
strike(r3,f3)
subject(strike,r3)
action(strike,s3)
 plural(s3)
 tense(s3,present)
 aircraft(f3)
```

26. [B]About to come under attack by planes of the US Navy in November 1943, Japanese warships maneuver frantically out of the harbor at Rabaul to head for open seas.

No results.

27. ᴳShips burning and sinking litter the harbor, while smoke and flames rise from battered shore installations and warehouses which were bombed during the 75 minute engagement.

My interpretation is:

plural(b2)
ship(b2)
vehicle(b2)
burn(b2)
action(burn,c2)
tense(c2,present)
transitive(c2)
and([b2,e2],j2)
sink(e2)
action(sink,f2)
tense(f2,present)
transitive(f2)
litter(j2,m2)
subject(litter,j2)
action(litter,k2)
plural(k2)
tense(k2,present)
definite(m2)
harbor(m2)
and([b2,q2],h5)
smoke(q2)
and([q2,s2],o3)
plural(s2)
flame(s2)
subject(rise,o3)
action(rise,a2)
plural(a2)
tense(a2,present)
intransitive(a2)
source(o3,r2)
state(r2,damaged)
shore(r2)
plural(r2)
installation(r2)
and([r2,t2],y2)
plural(t2)
warehouse(t2)
clshead(o3,which)
tense(o3,past)
plural(o3)
subject(bomb,g3)
action(bomb,h3)
transitive(h3)
time_spec(o3,n3)
definite(n3)
amount(n3)
cardinality(n3,75)
time_loc(n3)
engagement(n3)

28. ᴴPlanes from the aircraft carrier Saratoga raided the enemy naval base on New Britain Island and several of the 25 ships there were hit and set afire as they tried to reach the open sea.

My interpretation is:

plural(e2)
plane(e2)
source(e2,j2)
definite(j2)
aircraft(j2)
carrier(j2)
vehicle(j2)
name(j2,Saratoga)
raid(e2,e6)
subject(raid,e2)
action(raid,f2)
tense(f2,past)
definite(e6)
enemy(e6)
military(e6,navy)
base(e6)
time_spec(e6,v2)
name(v2,New Britain Island)
place(v2)
and([e2,w7],p18)
pronoun(w7,several)
subtype(w7,h3)
definite(h3)
amount(h3)
cardinality(h3,25)
plural(h3)
ship(h3)
vehicle(h3)
location(h3)
tense(w7,subjunctive)
singular(w7)
hit(u7)
action(hit,v7)
transitive(v7)
and([w7,h8],m18)
set(h8)
action(h8,burn)
coagent(h8,b8)
perprn(b8,they)
subject(try,h8)
action(try,v8)
tense(v8,past)
transitive(v8)
infinmarker(h8)
subject(reach,h8)
action(reach,b8)
plural(b8)
tense(b8,present)
transitive(b8)
definite(h8)
unlocked(h8)
purpose(h8)
sea(h8)
geofeature(h8)

29. [I]From October through December, 1943, Allied air forces inflicted such destruction at Rabaul on ships, planes, and installations that the Japanese abandoned it as a major base.

No results

30. [M]Circling wildly, the carrier Soryu is attacked by American dive bombers from Midway as her Zeros try in vain to defend her.

No results.

APPENDIX G
SOURCE CODE

This appendix provides the source code for the Captioned Based Interface. The code is currently located on the Computer Science ai9 computer in the directory ai9/work/dulle/CBI which is an open directory. To start the program from this directory you must start-up C-Prolog and then give the file **[nlp]** to load the program. After the program is loaded then issue the command **state** and the program will be ready to run.

The following code is blocked out in such away as to provide credit as to who wrote which sections.

```
/* parser16 */
/************************************************************************

The following section was written by Dr. Neil Rowe

************************************************************************/

/* This is top level of parser, with 2 inputs and 2 outputs: */
/* list to be parsed, part of speech it should be parsed as, */
/* the remaining list after the parse, and the meaning list generated. */

parse(L,P,L2,ML) :- abolish(cached_parse,4), abolish(cached_failed_parse,3),
  possible_parts_of_speech(L,PSL), parse2(L,P,[],PSL,L2,ML).

possible_parts_of_speech(L,PSL) :- bagof(PS,word_part_of_speech(L,PS),PSL).
word_part_of_speech(L,WPS) :- member(W,L), setof(PS,xd(W,PS),WPS).

/* The third and fourth arguments to parser2 are the stack of rules */
/* being used (to avoid infinite loops) and the possible parts of */
/* speech for each word, as an ordered sequence of sublists. */
```

```
parse2([],P,Stack,PSL,L2,ML) :- !, fail.
parse2(L,P,Stack,PSL,L2,ML) :- get_parse_rule(P,RPL,TPSL),
 not(member([P,RPL],Stack)), some_pair_intersect(PSL,TPSL),
 parse_items(L,RPL,[[P,RPL]|Stack],PSL,L2,ML).

parse_items(L,[RP],Stack,PSL,L2,ML) :- parse_item(L,RP,Stack,PSL,L2,ML).
parse_items(L,[RP1,RP2],Stack,PSL,L2,ML) :-
  parse_item(L,RP1,Stack,PSL,L3,ML1), length(L3,NL3),
  cut_to_length(PSL,NL3,PSL2), parse_item(L3,RP2,[],PSL2,L2,ML2),
  combine_meanings(ML1,ML2,RP1,RP2,ML), check_semantics(ML,RP1,RP2).

parse_item(L,RP,Stack,PSL,L2,ML) :- atomic_part_of_speech(RP), !,
 parse_item2(L,RP,Stack,PSL,L2,ML).
parse_item(L,RP,Stack,PSL,L2,ML) :- cached_parse(L,RP,L2,ML).
parse_item(L,RP,Stack,PSL,L2,ML) :- cached_parse(L,RP,L2,ML), !, fail.
parse_item(L,RP,Stack,PSL,L2,ML) :- cached_failed_parse(L,RP,L2), !, fail.
parse_item(L,RP,Stack,PSL,L2,ML) :- parse_item2(L,RP,Stack,PSL,L2,ML),
 add_cached_parse(L,RP,L2,ML).
parse_item(L,RP,Stack,PSL,L2,ML) :- not(cached_parse(L,RP,L2,ML)),
 not(cached_failed_parse(L,RP,L2)),
 asserta(cached_failed_parse(L,RP,L2)), !, fail.
add_cached_parse(L,RP,L2,ML) :- cached_parse(L,RP,L2,ML), !.
add_cached_parse(L,RP,L2,ML) :- asserta(cached_parse(L,RP,L2,ML)), !.

/* Fail if input multi-word and looking for only one word to cover all input */
/* (assumption: proper nouns are the only multi-word dictionary entries) */
parse_item2([X,Y|WL],RP,Stack,PSL,L,ML) :- atomic_part_of_speech(RP),
 not(var(L)), emptylist(L), not(RP=propnoun), !, fail.

/* When most words are parsed, a new variable is generated for them, */
/* which is put as the last argument to any meaning-list facts for the word */
parse_item2([X|L],RP,Stack,PSL,L,ML) :- atomic_part_of_speech(RP),
not(RP=verb),
 d(X,RP,ML2), get_variable(V), addvar(V,ML2,ML).

parse_item2([P|L],P,Stack,PSL,L,[]) :- punctuation(P), !.
parse_item2([X|L],P,Stack,PSL,L,[]) :- punctuation(P), !, fail.

/* Note verbs and participles have two separate meaning-list items: */
/* one with the verb name as predicate (to link direct objects of the verb), */
/* one with predicate "action" (to name the action, to link adverbial stuff) */

parse_item2([X|L],verb,Stack,PSL,L,[P,action(CX,V2)|ML4]) :-
```

```
canonical_verb(X,CX,ML1), d(CX,verb,ML2), append(ML1,ML2,ML3),
 get_variable(V), P=..[subject,CX,V], get_variable(V2), addvar(V2,ML3,ML4).


parse_item2([X|L],participle,Stack,PSL,L,[P,action(UX,V2)|ML3]) :-
 unparticiple(X,UX,ML1), d(UX,verb,ML2), union(ML1,ML2,ML12),
 get_variable(V), P=..[UX,V], get_variable(V2), addvar(V2,ML12,ML3).

/* A noun ending in "s" not in the dictionary is assumed plural */
parse_item2([X|L],noun,Stack,PSL,L,ML) :- name(X,AX), unplural(X,SX),
 d(SX,noun,ML2), get_variable(V), addvar(V,[plural|ML2],ML).

/* Single-letter words can be shape descriptors, either nouns or adjectives */
parse_item2([X|L],PS,Stack,PSL,L,[form(V,X)]) :- single_letter(X),
 (PS=noun;PS=adjective), get_variable(V), not(X=a).

/* Words with hyphens can be adverb-participle combinations */
parse_item2([X|L],adjl,Stack,PSL,L,ML) :- name(X,AX), ap-
pend(AX1,[45|AX2],AX),
 name(X1,AX1), name(X2,AX2), possible_parts_of_speech([X1,X2],PSL2),
 parse_item2([X2,X1],prtp,[],PSL2,[],ML).

/* Or words with hyphens can be adjective-noun adjectival combinations */
parse_item2([X|L],adjl,Stack,PSL,L,[subtype(V2,V)|ML]) :- name(X,AX),
 append(AX1,[45|AX2],AX), name(X1,AX1), name(X2,AX2),
 possible_parts_of_speech([X1,X2],PSL2),
 parse_item2([X1,X2],np,[],PSL2,[],ML), find_first_variable(ML,V),
 get_variable(V2).

/* Or words with hyphens can be noun-noun adjectival combinations */
/*
parse_item2([X|L],noun,Stack,PSL,L,[subtype(V2,V)|ML]) :- name(X,AX),
 append(AX1,[45|AX2],AX), name(X1,AX1), name(X2,AX2),
 possible_parts_of_speech([X1,X2],PSL2),
 parse_item2([X1,X2],noun,[],PSL2,[],ML), find_first_variable(ML,V),
 get_variable(V2).
*/
/* Or words with hyphens can be noun-verb adjectival combinations */
/*
parse_item2([X|L],noun,Stack,PSL,L,[subtype(V2,V)|ML]) :- name(X,AX),
 append(AX1,[45|AX2],AX), name(X1,AX1), name(X2,AX2),
 possible_parts_of_speech([X1,X2],PSL2),
 parse_item2([X1,X2],verb,[],PSL2,[],ML), find_first_variable(ML,V),
 get_variable(V2).
*/
/* Words with apostrophe-S are possessives and act like adjectives */
parse_item2([X|L],adjective,Stack,PSL,L,[owned_by(V2,V)|ML]) :- name(X,AX),
 append(AX1,[39,115],AX), name(X1,AX1), member(RP,[noun,propnoun,month]),
 d(X1,RP,ML2), get_variable(V), addvar(V,ML2,ML), get_variable(V2).

parse_item2(L,propnoun,Stack,PSL,L4,ML) :- first(L,FL), capitalized(FL),
 propnoun(L3,ML2), list(L3), append(L3,L4,L), get_variable(V), ad-
dvar(V,ML2,ML).

/* Otherwise, if word not in the dictionary, and not a simple */
/* grammatical category, it must be a grammar term */
parse_item2(L,RP,Stack,PSL,L2,ML) :- not(atomic_part_of_speech(RP)),
 parse2(L,RP,Stack,PSL,L2,ML).


/* Dictionary is stored with part of speech as predicate name; */
/* dictionary lookups must convert into proper expression form */
d(W,PS,ML) :- atomic_part_of_speech(PS), Q=..[PS,W,ML], call(Q).


/***********************************************************************

The following section was written by John Dulle and modified by Dr. Rowe.

***********************************************************************/

/* Next are the context-free grammar rules (parse tree must be binary) */
/* Note parse rules two or three arguments: left side, right side, */
/* and (optionally) some parts of speech necessary for rest of sentence, */
/* in order (if the there are options, they can be in sublists) */

get_parse_rule(PS,PSL,TPS) :- parse_rule(PS,PSL,TPS).
get_parse_rule(PS,PSL,[]) :- parse_rule(PS,PSL).

/* parse_rule(ng,[g]).
parse_rule(g,[dictg]). */

parse_rule(caption,[np]).

parse_rule(snt,[snt,period],
 [[noun,propnoun,pronoun,geo,month],
 [verb,tobe],
 period]).
parse_rule(snt,[snt,cj_snt],
 [[noun,propnoun,pronoun,geo,month],
 [verb,tobe],
 conjunction,
 [noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(snt,[cj_snt],
 [conjunction,
 [noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(snt,[cls_c,snt],
 [clausehead,
 [verb,tobe],
 comma,
 [noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(snt,[snt,cls],
 [[noun,propnoun,pronoun,geo,month],
 [verb,tobe],
 clausehead,
 [verb,tobe]]).
parse_rule(snt,[prtp_c,snt],
 [participle,
 comma,
 [noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(snt,[snt,c_prtp],
 [[noun,propnoun,pronoun,geo,month],
 [verb,tobe],
 comma,
 participle]).
parse_rule(snt,[snt,aps],
 [[noun,propnoun,pronoun,geo,month],
 [verb,tobe],
 [comma,openpar],
 [noun,propnoun,pronoun,geo,month],
 [comma,closepar]]).
parse_rule(snt,[snt,c_dt],
 [[noun,propnoun,pronoun,geo,month],
 [verb,tobe],
 comma,
 [month,numeric]]).
parse_rule(snt,[pps_c,snt],
 [preposition,
 [noun,propnoun,pronoun,geo,month],
 comma,
 [noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(snt,[pps,snt],
 [preposition,
 [noun,propnoun,pronoun,geo,month],
 [noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(snt,[np,vp],
 [[noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(cj_snt,[conjunction,snt],
 [conjunction,
 [noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(cj_snt,[comma,snt],
 [comma,
 [noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(cj_snt,[comma_cj,snt],
 [comma,
 conjunction,
 [noun,propnoun,pronoun,geo,month],
 [verb,tobe]]).
parse_rule(comma_cj,[comma,conjunction]).

parse_rule(np,[determiner,ng]).
parse_rule(np,[ng]).
parse_rule(np,[np,cj_np],
 [[noun,propnoun,pronoun,geo,month],
 conjunction,
 [noun,propnoun,pronoun,geo,month]]).
parse_rule(np,[np,ip],
 [[noun,propnoun,pronoun,geo,month],infinmarker,[verb,tobe]]).
parse_rule(np,[ip_c,ng],
 [infinmarker,[verb,tobe],[noun,propnoun,pronoun,geo,month]]).
parse_rule(np,[np,pps],
 [[noun,propnoun,pronoun,geo,month],
 preposition,
 [noun,propnoun,pronoun,geo,month]]).
parse_rule(np,[np,prtp],
 [[noun,propnoun,pronoun,geo,month],
 participle]).
parse_rule(np,[prtp_c,np],
 [participle,comma,[noun,propnoun,pronoun,geo,month]]).
parse_rule(np,[np,aps],
 [[noun,propnoun,pronoun,geo,month],
 [comma,openpar],
 [noun,propnoun,pronoun,geo,month],
 [comma,closepar]]).
parse_rule(cj_np,[comma,ng],
  [comma,
  [noun,propnoun,pronoun,geo,month]]).
parse_rule(cj_np,[comma_cj,np],
  [comma,
  conjunction,
  [noun,propnoun,pronoun,geo,month]]).
parse_rule(cj_np,[conjunction,np],
  [conjunction,
  [noun,propnoun,pronoun,geo,month]]).

parse_rule(ng,[noun],[noun]).
parse_rule(ng,[pronoun],[pronoun]).
parse_rule(ng,[geo],[propnoun]).
parse_rule(ng,[dt],[month]).
parse_rule(ng,[propnoun],[propnoun]).
parse_rule(ng,[adjl,ng]).

parse_rule(adjl,[adj]).
parse_rule(adjl,[adj,adjl]).
parse_rule(adjl,[adjl,adj]).
parse_rule(adjl,[adj,cj_adjl],[conjunction]).
parse_rule(cj_adjl,[conjunction,adj]).
parse_rule(adj,[adjective]).
parse_rule(adj,[numeric],[numeric]).
parse_rule(adj,[ng]).

parse_rule(vp,[cp_vp],[[verb,tobe],conjunction,[verb,tobe]]).
parse_rule(vp,[vp,ip],[[verb,tobe],infinmarker,[verb,tobe]]).
parse_rule(vp,[vg,pps],
 [[verb,tobe],
 preposition,
 [noun,propnoun,pronoun,geo,month]]).
parse_rule(vp,[vg,np],[[verb,tobe],[noun,propnoun,pronoun,geo,month]]).
parse_rule(vp,[tobe,np]).
parse_rule(vp,[vg],[[verb,tobe]]).
parse_rule(cp_vp,[vp,cj_vp]).
parse_rule(cj_vp,[conjunction,vp]).

parse_rule(vg,[adverb,vg],[adverb]).
parse_rule(vg,[tobe,vg],[tobe,[verb,tobe]]).
parse_rule(vg,[aux,vg],[aux,[verb,tobe]]).
parse_rule(vg,[verb],[verb]).
parse_rule(vg,[tobe,participle],[tobe,participle]).
parse_rule(vg,[tobe,pps],[tobe,preposition]).
parse_rule(vg,[verb,adverb],[verb,adverb]).

parse_rule(pps,[pp,pp],[preposition,preposition]).
parse_rule(pps,[adverb,pp],[adverb,preposition]).
parse_rule(pps,[comma,pps_c],[comma,preposition,comma]).
parse_rule(pps,[comma,pps],[comma,preposition]).
parse_rule(pps,[pp],[preposition]).
parse_rule(pps_c,[pps,comma],[preposition,comma]).
parse_rule(pp,[preposition,np],
 [preposition,
 [noun,propnoun,pronoun,geo,month]]).
parse_rule(pp,[preposition,numeric],
 [preposition,
 numeric]).

parse_rule(prtp,[comma,prtp_c],[comma,participle,comma]).
parse_rule(prtp,[prtp_c],[participle,comma]).
parse_rule(prtp,[b_prtp,pps],[participle,preposition]).
parse_rule(prtp,[b_prtp,cj_b_prtp],[participle,conjunction,participle]).
parse_rule(prtp,[b_prtp]).

parse_rule(prtp_c,[prtp,comma],[participle,comma]).
parse_rule(c_prtp,[comma,b_prtp],[comma,participle]).
parse_rule(cj_b_prtp,[conjunction,b_prtp],[conjunction,participle]).

parse_rule(b_prtp,[adverb,participle]).
parse_rule(b_prtp,[participle,adverb]).
parse_rule(b_prtp,[participle]).

parse_rule(ip,[adverb,ip],[adverb,infinmarker,[verb,tobe]]).
parse_rule(ip,[ip,np],
  [infinmarker,
  [verb,tobe],
  [noun,propnoun,pronoun,geo,month]]).
parse_rule(ip,[ip,pps],[infinmarker,[verb,tobe],preposition]).
parse_rule(ip,[pps,ip],[preposition,infinmarker,[verb,tobe]]).
parse_rule(ip_c,[ip,comma],[infinmarker,[verb,tobe],comma]).
parse_rule(ip,[infinmarker,vg]).

parse_rule(dt,[dt2,c_yr]).
parse_rule(dt,[dt2]).
parse_rule(dt2,[month,numeric]).
parse_rule(dt2,[month]).
parse_rule(c_yr,[comma,numeric]).
parse_rule(c_dt,[comma,dt]).

parse_rule(c_geo,[comma,geo]).
parse_rule(geo,[propnoun]).
parse_rule(geo,[propnoun,c_geo]).

parse_rule(cls,[clausehead,snt]).
parse_rule(cls,[clausehead,vp]).
parse_rule(cls_c,[cls,comma]).

parse_rule(aps,[openpar,apb],[openpar]).
parse_rule(aps,[c_np],[comma]).
parse_rule(aps,[c_dt],[comma]).
parse_rule(aps,[c_np,comma],[comma,comma]).
parse_rule(aps,[c_dt,comma],[comma,comma]).
parse_rule(apb,[np,closepar]).
parse_rule(c_np,[comma,np]).
parse_rule(snt,[doesword,snt]).


**************************************************************************

The following section was written by Dr. Neil Rowe, Gene Guglielm, and John Dulle

**************************************************************************/
/*
"combine_meanings" does the tricky stuff in semantic interpretation:
it handles two-term parse rules, combining the meaning lists for the
parses of the two terms.
```

```
*/

combine_meanings(ML1,ML2,preposition,np,ML) :-
    singlemember(property(Q,V),ML1),
    !,
    find_first_variable(ML2,Y),
    get_variable(X),
    P=..[V,X,Y],
    delete(property(Q,V),ML1,ML1X),
    union([P|ML1X],ML2,ML),
    !.

/*
Rule to tie together the subject and object when encountering a transitive
verb.
*/

combine_meanings(ML1,ML2,vg,np,ML) :-
    singlemember(transitive(VerbVar),ML1),
    !,
    singlemember(subject(Verb,WhoVar),ML1),
    find_first_variable(ML2,ObjVar),
    PX=..[Verb,WhoVar,ObjVar],
    delete(transitive(VerbVar),ML1,ML1X),
    union([PX|ML1X],ML2,ML),
    !.
/*
Rule adjusts for the tenses between the aux/tobe and the verb/participle.
*/
combine_meanings(ML1,ML2,S1,S2,ML) :-
    member([S1,S2],[[aux,vg],[tobe,vg],[tobe,participle]]),
    !,singlemember(tense(_,TA),ML1),
    delete(tense(_,TV),ML2,ML2X),
    union(ML1,ML2X,ML),
    !.
combine_meanings(ML1,ML2,S1,S2,ML) :-
    member([S1,S2],[[aux,vg],[tobe,vg],[tobe,participle]]),
    !,singlemember(tense(_,TA),ML1),
    not(singlemember(tense(_,TV),ML2)),
    singlemember(action(X,Y),ML2),
    union([tense(X,TA)],ML2,ML),
    !.
combine_meanings(ML1,ML2,S1,S2,ML2) :-
```

```
    member([S1,S2],[[aux,vg],[tobe,vg],[tobe,participle]]),
    !.

combine_meanings(ML1,ML2,S1,S2,ML) :-
    member([S1,S2],[[determiner,ng],[adjl,ng],[adverb,vg]]),
    !,
    find_first_variable(ML1,X),
    find_first_variable(ML2,Y),
    change_variable(X,Y,ML1,ML1X),
    union(ML1X,ML2,ML),
    !.

combine_meanings(ML1,ML2,S1,S2,ML) :-
    member([S1,S2],[[doesword,sentence]]),
    !,
    find_first_variable(ML1,X),
    singlemember(action(A,Y),ML2),
    change_variable(X,Y,ML1,ML1X),
    delete(tense(Y,T),ML2,ML2X),
    delete(plural(Y),ML2X,ML2Y),
    union(ML2Y,ML1X,ML),
    !.

combine_meanings([P|ML1],ML2,participle,np,ML) :-
    !,
    find_first_variable(ML2,Y),
    P=..[A,B],
    PX=..[A,B,Y],
    union([PX|ML1],ML2,ML),
    !.

combine_meanings(ML1,ML2,clausehead,snt,ML) :-
    !,
    singlemember(clshead(Q,V),ML1),
    find_first_variable(ML2,Y),
    get_variable(X),
    P=..[V,X,Y],
    delete(clshead(Q,V),ML1,ML1X),
    union([P|ML1X],ML2,ML),
    !.

combine_meanings(ML1,ML2,S1,S2,ML) :-
    member([S1,S2],[[adjl,cj_adjl],[vp,cj_vp],[b_prtp,cj_b_prtp],
    [snt,cj_snt],[np,cj_np]]),
    !,
    find_first_variable(ML1,X),
    member(conjunction(Z,C),ML2),
    delete(conjunction(Z,C),ML2,ML2X),
    find_first_variable(ML2X,Y),
    get_variable(V), P=..[C,[X,Y],V],
    union(ML1,[P|ML2X],ML),
    !.

combine_meanings(ML,[],snt,period,ML) :-
    !.

combine_meanings([],ML,comma,SP,ML) :-
    member(SP,[conjunction, snt, pps_c, pps, b_prtp, prtp_c,
    numeric, dt, geo, ng, np]),
    !.

combine_meanings(ML,[],SP,comma,ML) :-
    member(SP, [pps, prtp, cls, c_np, c_dt, ip]),
    !.

combine_meanings(ML,[],np,closepar,ML) :-
    !.

combine_meanings([],ML,openpar,apb,ML) :-
    !.

/*
Rule to handle multiple subjects performing a single action.
*/

combine_meanings(ML1,ML2,S1,S2,ML) :-
    member(and(_,X),ML1),
    !,
    find_first_variable(ML2,Y),
    change_variable(X,Y,ML1,ML1X), union(ML1X,ML2,ML),
    !.

/*
This is the default combined meanings.
*/
```

```
combine_meanings(ML1,ML2,S1,S2,ML) :-
    !,
    find_first_variable(ML1,X),
    find_first_variable(ML2,Y),
    change_variable(X,Y,ML1,ML1X),
    union(ML1X,ML2,ML),
    !.

check_list(MSG,X) :- nl,nl,write('Meaning List For:  '),
write(X),nl,check_list2(MSG).
check_list2(MSG) :- member(Y,MSG),write('    '),write(Y),nl,fail.
check_list2(MSG).
write_list(ML) :- member(Y,ML),write('    '),write(Y),nl,fail.
write_list(ML).



/* Running of semantic checks; this code could be greatly expanded */

check_semantics(ML,RP1,pp) :- !, not(case_violation(ML)), !.
check_semantics(ML,RP1,RP2) :- !.

case_violation(ML) :- member(P,[time_spec(A,B),in_period(A,B),
  location(A,B),inside(A,B),
  destination(A,B),source(A,B),purpose(A,B),beneficiary(A,B),coagent(A,B),
  tool(A,B),above(A,B),below(A,B)]),
  member(P,ML), P=..[Pred,A,B], case_violation2(Pred,A,B,ML).

*****************************************************************************

The following section was written by Dr. Neil Rowe, Dr. Holtkamp, and John Dulle

******************************************************************************/
/*
case_violation(ML) :- write('Check_semantics failed on '), write(ML), nl, !,fail.
*/

case_violation2(P,A,B,ML) :-
  member(P,[location,destination,source]), not(location_name(B,ML)).
case_violation2(P,A,B,ML) :-
  member(P,[time_spec,in_period,time_dest,time_src]), not(time_loc_name(B,ML)).
case_violation2(P,A,B,ML) :-
  member(P,[inside,above,location,source,destination,purpose,beneficiary]),
 time_loc_name(B,ML).
case_violation2(P,A,B,ML) :- member(P,[coagent]),
 location_name(A,ML).
case_violation2(P,A,B,ML) :- member(P,[purpose,beneficiary,coagent]),
 location_name(B,ML).

location_name(X,ML) :-
 member(P,[place,region,geofeature,river,junction,vegetation,plant,edge]),
 Q=..[P,X], member(Q,ML), !.
location_name(X,ML) :- member(road(X,V),ML), !.


time_loc_name(X,ML) :-
 member(P,[month_name,holiday_name,time_loc,action]),
 Q=..[P,X], member(Q,ML), !.



/* Morphology */

/* This tries to find the singular form of a plural noun */
unplural(PV,SV) :- name(PV,APV), name('s',AS), append(ASV,AS,APV),
 name(SV,ASV).



/* This tries to find the present-tense singular form of a given verb, */
/* with meaning list holding the meaning of the original suffix */


canonical_verb(V,CV,[singular,tense(present)]) :- name(V,AV), name('s',AD),
 append(ACV,AD,AV), name(CV,ACV).
canonical_verb(V,CV,[singular,tense(present)]) :- name(V,AV), name('es',AD),
 append(ACV,AD,AV), name(CV,ACV), last(ACV,AI), consanant(AI).
canonical_verb(V,CV,[tense(past)]) :- name(V,AV), name('d',AD),
 append(ACV,AD,AV), name(CV,ACV).
canonical_verb(V,CV,[tense(past)]) :- name(V,AV), name('ed',AD),
 append(ACV,AD,AV), name(CV,ACV), last(ACV,AI), consanant(AI).
canonical_verb(had,have,[tense(past)]).
canonical_verb(has,have,[singular,tense(present)]).
canonical_verb(came,come,[tense(past)]).
canonical_verb(went,go,[tense(past)]).
canonical_verb(left,leave,[tense(past)]).
canonical_verb(blew,blow,[tense(past)]).
canonical_verb(burnt,burn,[tense(past)]).
canonical_verb(cut,cut,[tense(past)]).
canonical_verb(dove,dive,[tense(past)]).
canonical_verb(fought,fight,[tense(past)]).
canonical_verb(got,get,[tense(past)]).
canonical_verb(hit,hit,[tense(past)]).
canonical_verb(knew,know,[tense(past)]).
canonical_verb(laid,lay,[tense(past)]).
canonical_verb(left,leave,[tense(past)]).
canonical_verb(made,make,[tense(past)]).
canonical_verb(met,meet,[tense(past)]).
canonical_verb(rose,rise,[tense(past)]).
canonical_verb(ran,run,[tense(past)]).
canonical_verb(saw,see,[tense(past)]).
canonical_verb(set,set,[tense(past)]).
canonical_verb(shot,shoot,[tense(past)]).
canonical_verb(sank,sink,[tense(past)]).
canonical_verb(sunk,sink,[tense(past)]).
canonical_verb(struck,strike,[tense(past)]).
canonical_verb(took,take,[tense(past)]).
canonical_verb(tried,try,[tense(past)]).

canonical_verb(V,V,[plural,tense(present)]).


/* This finds the verb corresponding to a participle */

unparticiple(X,UX,[tense(present)]) :- name(X,AX), name('ing',AING),
  append(AF,AING,AX), unparticiple2(AF,AUX), name(UX,AUX).
unparticiple(X,UX,[tense(past)]) :- name(X,AX), name('d',AD),
  append(AF,AD,AX), unparticiple2(AF,AUX), name(UX,AUX).
unparticiple(being,is,[tense(present)]).
unparticiple(been,is,[tense(past)]).
unparticiple(circling,circle,[tense(present)]).
unparticiple(run,run,[tense(past)]).
unparticiple(gone,go,[tense(past)]).
unparticiple(see,seen,[tense(past)]).
unparticiple(blown,blow,[tense(past)]).
unparticiple(burnt,burn,[tense(past)]).
unparticiple(cut,cut,[tense(past)]).
unparticiple(dove,dive,[tense(past)]).
unparticiple(got,get,[tense(past)]).
unparticiple(hit,hit,[tense(past)]).
unparticiple(known,knew,[tense(past)]).
unparticiple(led,lead,[tense(past)]).
unparticiple(left,leave,[tense(past)]).
unparticiple(made,make,[tense(past)]).
unparticiple(met,meet,[tense(past)]).
unparticiple(risen,rise,[tense(past)]).
unparticiple(seen,see,[tense(past)]).
unparticiple(set,set,[tense(past)]).
unparticiple(shot,shoot,[tense(past)]).
unparticiple(struck,strike,[tense(past)]).
unparticiple(sunk,sink,[tense(past)]).
unparticiple(taken,take,[tense(past)]).
unparticiple(wading,wade,[tense(present)]).

unparticiple2(AF,AF).
unparticiple2(AF,AUX) :- name('e',AE), append(AUX,AE,AF).
unparticiple2(AF,AUX) :- last(AF,LAF), butlast(AF,AUX), last(AUX,LAF).

*********************************************************************

The following section was written by Dr. Neil Rowe, Dr. Holtkamp.

*********************************************************************/
consanant(A) :- singlemember(A,[98,99,100,102,103,104,106,107,108,109,110,
  112,113,114,115,116,118,119,120,121,122]).

capitalized(W) :- name(W,AW), first(AW,FAW), FAW>64, FAW<91, !.

/* Manipulation of "variables" (which actually appear as small letters) */

parse_number(1).

get_variable(V) :- retract(variable_name(X)), name(X,AX), parse_number(N),
  name(N,AN), append(AX,AN,AV), name(V,AV), !.
get_variable(V) :- reconsult(variables), !, retract(parse_number(N)),
  Np1 is N+1, asserta(parse_number(Np1)), get_variable(V).

addvar(X,[],[]).
addvar(X,[P|L],[PX|PL]) :- addvar2(X,P,PX), addvar(X,L,PL).

addvar2(X,P,PX) :-  atom(P), PX=..[P,X].
addvar2(X,P,PX) :-  P=..[P2,Y], PX=..[P2,X,Y].
```

```
addvar2(X,P,PX) :-  P=..[P2,Y,Z], PX =..[P2,X,Y,Z]. /* bh */

find_first_variable([E|ML],X) :-  E=..[P,X|L], possible_variable_name(X), !.
find_first_variable([E|ML],Y) :-  E=..[P,X,Y|L], possible_variable_name(Y), !.
find_first_variable([E|ML],X) :-  find_first_variable(ML,X), !.

possible_variable_name([]) :- !, fail.
possible_variable_name([A|B]) :- !, fail.

possible_variable_name(X) :- not(var(X)), name(X,[AV|AN]), name(N,AN),
  number(N), AV>96, AV<123, !.

single_letter([A|B]) :- !, fail.
single_letter(X) :- not(var(X)), name(X,[A]), A>64, A<123, !.

change_variable(A,B,PL1,PL2) :- change_variable2(A,B,PL1,PL2),
  name(A,[AFA|ABFA]), name(FA,[AFA]), good_assert(variable_name(FA)), !.
change_variable2(A,B,[],[]).
change_variable2(A,B,[P|L],[P2|M]) :- P=..[R,A],
   !, P2=..[R,B], change_variable2(A,B,L,M).
change_variable2(A,B,[P|L],[P2|M]) :- P=..[R,A,C],
   !, P2=..[R,B,C], change_variable2(A,B,L,M).
change_variable2(A,B,[P|L],[P2|M]) :- P=..[R,C,A],
   !, P2=..[R,C,B], change_variable2(A,B,L,M).
change_variable2(A,B,[P|L],[P|M]) :- !, change_variable2(A,B,L,M).

good_assert(X) :- call(X), !.
good_assert(X) :- assertz(X), !.

/* I/O stuff */

ask :- abolish(meaning,1),
 write('Type your question in English (no capitals or punctuation, please):'),
 nl, readlineclump(L), write_parse(L,question).

carefulstate :- abolish(meaning,1), carefulstate2, bagof(M,meaning(M),ML),
  resolve_references(ML,ML2), write('Final meaning list is:'), nl,
  write(ML2), nl.
carefulstate2 :-
  write('Type your partial caption in English (no capitals or punctuation,'),
  nl, write('please), or empty line to stop:'),
  nl, readlineclump(L), carefulstate3(L).
carefulstate3([]) :- !.
```

```
carefulstate3(L) :- careful_write_parse(L,caption), carefulstate2.

careful_write_parse(L,NAME) :- write(L), nl, not(badword(L,W)),
  bagof(ML,parse(L,NAME,[],ML),MLL), write_meanings(MLL),
  first(MLL,FML), assertz(meaning(FML)), !.
careful_write_parse(L,NAME) :- write('I cannot parse that.'), nl.

state :- initialize, state2, bagof(M,meaning(M),ML),
  resolve_references(ML,ML2), write('Final meaning list is:'), nl,
  write(ML2), nl.
state2 :- nl, write('Type your partial caption in English'),
  nl, write('empty line to stop:'),
  nl, readlineclump(L), fix_first_letter(L,L2), write(L2), nl, state3(L2).

initialize :- abolish(meaning,1), abolish(variable_name,1), abolish(parse_number,1),
asserta(parse_number(1)).

fix_first_letter(WL,WL) :- propnoun(WL2,_), append(WL2,_,WL), !.
fix_first_letter([W|WL],[W|WL]) :- name(W,[N1|NL]), N1>96, !.
fix_first_letter([W|WL],[W|WL]) :- d(W,_,_), !.
fix_first_letter([W|WL],[NW|WL]) :- name(W,[N1|NL]), N2 is N1+32,
  name(NW,[N2|NL]), !.

state3([]) :- !.
state3(L) :- last(L,'period'), write_parse(L,snt), state2.
state3(L) :- not(last(L,'period')),  write_parse(L,caption), state2.

write_parse(L,NAME) :- not(badword(L,W)), parse(L,NAME,[],ML),
  write_meaning(ML), assertz(meaning(ML)), !.

/* kmw - version
write_parse(L,NAME) :- write(L), nl, (badword(L,W), !, fail;
  parse(L,NAME,[],ML), assertz(meaning(ML)), !).
 end kmw -version */
write_parse(L,NAME) :- nl, nl,
                write('!! structure error !!'),
                nl, nl, !, state2.

p(L) :-  abolish(meaning,1), write_parse(L,caption), !.


write_meaning(ML) :- nl, write('My interpretation is:'), nl, write_list(ML), nl.
write_meanings([ML]) :- write('Only interpretation is:'), nl, write(ML), nl.
write_meanings(MLL) :- write('Possible interpretations (first is preferred):'),
  nl, write_list(MLL).

/* A word not in dictionary nor parse rules is an error */
badword(L,W) :- member(W,L), not(xd(W,PS)), write('word error:'), write(W), nl, !.

/* "Extended dictionary"--confirms if something is word */
xd(W,RP) :- d(W,RP,ML).
xd(W,numeric) :- numeric(W,ML), !.
xd(W,propnoun) :- capitalized(W),propnoun(WL,RP), member(W,WL), !.
xd(W,adjective) :- name(W,AW), append(AW1,[39,115],AW), name(W1,AW1),
  (xd(W1,noun); xd(W1,propnoun); xd(W1,month)), !.
xd(W,noun) :- unplural(W,W2), d(W2,noun,ML2).
xd(W,verb) :- canonical_verb(W,W2,ML), d(W2,verb,ML2).
xd(W,participle) :- unparticiple(W,W2,ML), d(W2,verb,ML2).
xd(W,noun) :- name(W,[A]), not(number(W)).
xd(W,adjective) :- name(W,AW), append(AW1,[45|AW2],AW), name(W1,AW1),
  name(W2,AW2), xd(W1), xd(W2).
xd(W,W) :- punctuation(W).

punctuation(W) :- member(W,[period,comma,openpar,closepar,dash,semicolon]).

atomic_part_of_speech(X) :- member(X,[propnoun,noun,verb,adjective,determiner,
  preposition,numeric,tobe,aux,adverb,period,comma,openpar,closepar,dash,
  semicolon,conjunction,month,pronoun,clausehead,infinmarker,doesword]).

/* Anaphoric references */

/* This changes definite references (things in English preceded by "the") */
/* to the most recent noun of that kind. */

resolve_references(ML,ML2) :- resolve_references2(ML,ML3),
  delete_funny_preds(ML3,ML2), !.
resolve_references2([],[]) :- !.
resolve_references2([ML],ML) :- !.
resolve_references2([ML1,ML2|ML],OML) :- member(definite(X),ML2),
  member(M,ML2), M=..[P,X], not(special_type(P)), M2=..[P,Y],
  member(M2,ML1), not(X=Y), not(better_type_pred(P,X,ML2)),
  !, delete(definite(X),ML2,ML2a), change_variable(X,Y,ML2a,ML2b),
  resolve_references2([ML1,ML2b|ML],OML).
resolve_references2([ML1,ML2|ML],OML) :- member(definite(X),ML2),
  member(M,ML2), M=..[P,X,C], not(variable_name(C)), M2=..[P,Y,C],
  member(M2,ML1), not(X=Y), !, delete(definite(X),ML2,ML2a),
 change_variable(X,Y,ML2a,ML2b), resolve_references2([ML1,ML2b|ML],OML).
resolve_references2([ML1,ML2|ML],OML) :- !, union(ML1,ML2,ML12),
 resolve_references2([ML12|ML],OML).

better_type_pred(P,X,ML) :- member(P,[region,geofeature,terrain]),
 not(special_type(P)), member(M3,ML), M3=..[Q,X],
 not(member(Q,[region,geofeature,terrain])), not(special_type(Q)).

delete_funny_preds([],[]).

delete_funny_preds([P|PL],NPL) :- P=..[Q,A], funny_type(Q), !,
 delete_funny_preds(PL,NPL).

delete_funny_preds([P|PL],[P|NPL]) :- delete_funny_preds(PL,NPL).

funny_type(P) :- member(P,[definite,indefinite]).
special_type(P) :- member(P,[definite,indefinite,plural]).

/* number processing -bh- */

numeric(W,ML) :- cardinal(W,Cval),
 append([amount],[cardinality(Cval)],ML).

numeric(W,ML) :- ordinal(W,Oval),
 append([rank],[cardinality(Oval)],ML).

cardinal(C,Cval) :- name(C,X),int(X),name(Cval,X).
cardinal(C,Cval) :- name(C,X),floating(X),name(Cval,X).
cardinal(C,Cval) :- name(C,X),fraction(X),name(Cval,X).

ordinal(O,Oval)  :- name(O,X), last(X,LL),butlast(X,X1),
 last(X1,BL),butlast(X1,X2), int(X2), append([BL],[LL],Ord),
 ord(Ord),name(Oval,X2).
ord(OE) :- OE == [110,100].
ord(OE) :- OE == [114,100].
ord(OE) :- OE == [115,116].
ord(OE) :- OE == [116,104].

int(X) :- check_digit(X).

floating(X) :- singlemember(46,X),left_side(X,46,[],Left,Right),
 check_digit(Left), check_digit(Right).
```

```
fraction(X) :- singlemember(47,X),left_side(X,47,[],Left,Right),
  check_digit(Left), check_digit(Right).

check_digit([]).
check_digit([Xf|Xr]) :- digit(Xf), check_digit(Xr).

left_side([],_,_,_,_).
left_side([Xf|Xr],T,XL,XL,Xr) :- Xf == T.
left_side(X,T,XL,Xl,Xr) :- first(X,Xf), append(XL,[Xf],XL2),
  tail(X,XR), left_side(XR,T,XL2,Xl,Xr).

digit(D) :- singlemember(D,[48,49,50,51,52,53,54,55,56,57]).

/* Input reading */

readlineclump(L) :- niceread(S), clumpstring(S,AL), makenames(AL,L), !.

makenames([],[]).
makenames([AL|LL],[NL|NLL]) :- name(NL,AL), makenames(LL,NLL).

clumpstring(L3,[L1|L5]) :- nextclump(L3,[],L1,L2), !, clumpstring(L2,L5).
clumpstring(L,[L]) :- member(X,L), not(terminator(X)), !.
clumpstring(L,[]).

nextclump([],L1,RL1,[]) :- not(L1=[]), !, reverse(L1,RL1).
nextclump([44|L],[],CL,L) :- !, name('comma',CL).
nextclump([44|L],L1,RL1,[44|L]) :- !, reverse(L1,RL1).
nextclump([46|L],[],CL,L) :- !, name('period',CL).
nextclump([46|L],L1,RL1,[46|L]) :- !, reverse(L1,RL1).
nextclump([40|L],[],CL,L) :- !, name('openpar',CL).
nextclump([40|L],L1,RL1,[40|L]) :- !, reverse(L1,RL1).
nextclump([41|L],[],CL,L) :- !, name('closepar',CL).
nextclump([41|L],L1,RL1,[41|L]) :- !, reverse(L1,RL1).
nextclump([T|L],[],L2,L3) :- terminator(T), !, nextclump(L,[],L2,L3).
nextclump([T|L],L1,RL1,L) :- terminator(T), !, reverse(L1,RL1).
nextclump([X|L],L1,L2,L3) :- nextclump(L,[X|L1],L2,L3).

terminator(9).
terminator(32).
terminator(58).
terminator(59).

niceread(L) :- checkretract(readbuff(L2)), asserta(readbuff([])), niceread2(L), !.
```

```
niceread2(L) :- get0(C), niceread3(C,L).
niceread3(10,L) :- !, readbuff(L2), reverse(L2,L).
niceread3(C,L) :- readbuff(L3), retract(readbuff(L3)),
  asserta(readbuff([C|L3])), niceread2(L).

checkretract(S) :- call(S), retract(S), !.
checkretract(S).

/* List processing */

first([X|L],X).

last([X],X) :- !.
last([X|L],Y) :- last(L,Y).

tail([X|Tail],Tail).

emptylist([]).

butlast(L,M) :- append(M,[X],L).

union([],L,L).
union([X|L],L2,L3) :- singlemember(X,L2), !, union(L,L2,L3).
union([X|L],L2,[X|L3]) :- union(L,L2,L3).

some_pair_intersect(LL,[]).
some_pair_intersect(LL,[I2|L2]) :- list(I2), !, append(LL1,[L|LL2],LL),
  intersect(L,I2), !, some_pair_intersect(LL2,L2).
some_pair_intersect(LL,[I2|L2]) :- append(LL1,[L|LL2],LL),
  member(I2,L), !, some_pair_intersect(LL2,L2).

intersect(L1,L2) :- member(X,L1), member(X,L2), !.

member(X,[X|L]).
member(X,[Y|L]) :- member(X,L).

singlemember(X,[X|L]) :- !.
singlemember(X,[Y|L]) :- singlemember(X,L).

delete(X,[],[]).
delete(X,[X|L],M) :- !, delete(X,L,M).
delete(X,[Y|L],[Y|M]) :- delete(X,L,M).
```

```
append([],L,L).
append([X|L],L2,[X|L3]) :- append(L,L2,L3).

reverse(L,R) :- reverse2(L,[],R).
reverse2([],L,L) :- !.
reverse2([X|L],R,S) :- reverse2(L,[X|R],S).

cut_to_length(L,N,L2) :- append(L1,L2,L), length(L2,N), !.

list([A|B]).
list([]).

write_list([]).
write_list([L|LL]) :- write(L), write(' '), write_list(LL).


/* Routine to take a meaning list and write it into a file */
/* in the form of a rule, with true variables (upper case) substituted */
/* and the first argument changed to the picture number described */

writefacts(ML,File,Picture) :- tell(File), writefacts2(ML,Picture), told.
writefacts2([],N).
writefacts2([M|ML],N) :- writefacts3(M,N), write('.'), nl, writefacts2(ML,N).
writefacts3(M,N) :- atom(M), MW=..[M,N], writeq(MW).
writefacts3(M,N) :- M=..[P,A], MW=..[P,N,A], writeq(MW).
writefacts3(M,N) :- M=..[P,A,B], MW=..[P,N,A,B], writeq(MW).

/* Routine to find a picture description that contains something matching */
/* a given natural-language query. It first parses the (possibly multiline) */
/* query into a meaning list. It then translates the pseudo-variables in */
/* the meaning list into true variables (by the trick of writing the */
/* pseudo-variables with initial capitals into a file, then loading file). */
/* It then executes the list of expressions with variables as a query of */
/* conjunctive terms, and returns as answer the filled-in query. */
/* (Picture-description files must be loaded before this program is run.) */

query(File,Id) :- abolish(meaning,1), write('type your questions :'), nl, state2,
  bagof(M,meaning(M),ML), resolve_references(ML,ML2),
  translate_query(ML2,File), query_pred(Id).

query_pred(Id) :- query_pred1(Id1), write(Id), write(' = '), write(Id1), nl.

queryc(File) :- abolish(meaning,1), write('type your questions :'), nl, state2,
 bagof(M,meaning(M),ML), resolve_references(ML,ML2),
 translate_query(ML2,File).

/* This following is required because Prolog does not send the bindings
  to stdout! */

query_predc(Id) :- query_pred1(Id1), write(Id), write(' = '), write(Id1), nl,
 get0(C), fail.

translate_query(ML,File) :- tell(File),
 write('query_pred1(Id) :- '), translate_query2(ML), write('.'), nl,
 told, reconsult(File).
translate_query2([M]) :- translate_term(M), !.
translate_query2([M|ML]) :- translate_term(M), write(','), nl,
 translate_query2(ML).

translate_term(M) :- atom(M), write(M), write('(Id)'), !.
translate_term(M) :- M=..[P,A], variablize(A,VA), write(P),
 write('(Id,'), write(VA), write(')'), !.
translate_term(M) :- M=..[P,A,B], variablize(A,VA), variablize(B,VB),
 write(P), write('(Id,'), write(VA), write(','), write(VB), write(')'), !.


variablize(S,VS) :- possible_variable_name(S), name(S,[AL|AN]),
 upper_case_ascii(AL,AU), name(VS,[AU|AN]), !.
variablize([S],[VS]) :- variablize(S,VS).
variablize([S|SL],[VS|VSL]) :- possible_variable_name(S), name(S,[AL|AN]),
 upper_case_ascii(AL,AU), name(VS,[AU|AN]), variablize(SL,VSL), !.
variablize(S,S).

upper_case_ascii(AL,AU) :- AU is AL-32.

/* execute_list_query([]).

execute_list_query([M|ML]) :- call(M), execute_list_query(ML). */


/* introduce class hierarchies for nouns - bh - */

inherit_attr(ML,ML) :- not(member(superclasses(ID,Obj,ClassList),ML)), !.

inherit_attr(ML,MLout) :-
 member(superclasses(ID,Obj,ClassList),ML),
  delete(superclasses(ID,Obj,ClassList),ML,ML2),
  add_superclass_attr(ID,ClassList,ML2,ML3),
  inherit_attr(ML3,MLout).


add_superclass_attr(ID, [], ML, ML).

add_superclass_attr(ID,[Class|RClassList],ML,MLout) :-
  d(Class,noun,MLraw),
  addvar(ID,MLraw,MLsup),
  union(MLsup,ML,ML3),
  add_superclass_attr(ID,RClassList,ML3,MLout).

/* Prints the cached info */
pc :- listing(cached_parse), listing(cached_failed_parse).

stat(a).

same(A,A).
```

APPENDIX H

CURRENT DICTIONARY

This appendix provides the dictionary that was used to parse the captions. Each entry is classified by its part of speech and contains a meaning-list for each sense of the word.

```
adjective('Allied',[nationality(allied)]).
adjective('American',[nationality(us)]).
adjective('Japanese',[nationality(japan)]).
adjective('NATO',[nationality(nato)]).
adjective('Philippine',[nationality(philippine)]).
adjective('Russian',[nationality(ussr)]).
adjective('Soviet',[nationality(ussr)]).
adjective('US',[nationality(us)]).
adjective(american,[nationality(us)]).
adjective(bare,[terrain(unforested)]).
adjective(battered,[state(damaged)]).
adjective(big,[size(+)]).
adjective(black,[color(black)]).
adjective(british,[nationality(uk)]).
adjective(broad,[width(+)]).
adjective(bursting,[action(+)]).
adjective(clustered,[dispersion(narrow)]).
adjective(coastal,[terrain(coast)]).
adjective(constant,[action(speed(0))]).
adjective(crazy,[action(eratic)]).
adjective(decisive,[action(thought)]).
adjective(dive,[action(-)]).
adjective(down,[direction(-)]).
adjective(east,[direction(90)]).
adjective(eight,[cardinality(8)]).
adjective(evasive,[action(movement)]).
adjective(fast,[action(speed(+))]).
adjective(few,[cardinality(-)]).
adjective(first,[cardinality(1)]).
adjective(five,[cardinality(5)]).
adjective(flat,[height(0)]).
adjective(forested,[terrain(forested)]).
adjective(four,[cardinality(4)]).
adjective(frantic,[action(speed(+))]).
adjective(german,[nationality(germany)]).
adjective(gray,[color(gray)]).
adjective(heavy,[weight(+)]).
adjective(heterogeneous,[texture(rough)]).
adjective(high,[height(+)]).
adjective(homogeneous,[texture(smooth)]).
adjective(landlocked,[terrain(seashore)]).
adjective(large,[size(+)]).
adjective(left,[xcoordinate(-)]).
adjective(lone,[cardinality(1)]).
adjective(long,[size(+)]).    /* bh */
adjective(lower,[ycoordinate(-)]).
adjective(main,[main]).
adjective(major,[major]).
adjective(many,[cardinality(+)]).
adjective(middle,[xcoordinate(0),ycoordinate(0)]).
adjective(mixed,[texture(rough)]).
adjective(narrow,[width(-)]).
adjective(nato,[nationality(nato)]).
adjective(naval,[military(navy)]).
adjective(near,[distance(-)]).
adjective(nine,[cardinality(9)]).
adjective(no,[cardinality(0)]).
adjective(north,[direction(0)]).
adjective(northeast,[direction(45)]).
adjective(northwest,[direction(315)]).
adjective(one,[cardinality(1)]).
adjective(open,[unlocked,purpose]). /* bh */
adjective(other,[cardinality(1),different]).
adjective(pre,[prefix(+),before]).
adjective(reinforced,[action(+)]).
adjective(right,[xcoordinate(+)]).
adjective(rough,[texture(rough)]).
adjective(russian,[nationality(ussr)]).
adjective(scarce,[dispersion(wide)]).
adjective(scattered,[dispersion(wide)]).
adjective(second,[cardinality(1),different]).
adjective(separate,[cardinality(1),different]).
adjective(seven,[cardinality(7)]).
adjective(sharp,[sharp]).
adjective(short,[height(-)]).
adjective(similar,[similar]).
adjective(single,[cardinality(1)]).
adjective(six,[cardinality(6)]).
adjective(slim,[width(-)]).
adjective(small,[size(-)]).
adjective(smooth,[texture(smooth)]).
adjective(south,[direction(180)]).
adjective(southeast,[direction(135)]).
adjective(southwest,[direction(225)]).
adjective(soviet,[nationality(ussr)]).
adjective(strong,[strength(+)]).
adjective(such,[such]).
adjective(tall,[height(+)]).
adjective(three,[cardinality(3)]).
adjective(this,[this]).
adjective(tiny,[size(-)]).
adjective(two,[cardinality(2)]).
adjective(up,[direction(+)]).
adjective(upper,[ycoordinate(+)]).
adjective(us,[nationality(us)]).
adjective(various,[cardinality(+)]).
adjective(west,[direction(270)]).
adjective(western,[direction(270)]).
adjective(white,[color(white)]).
adjective(wide,[width(+)]).
adjective(wild,[action(wild)]).

adverb(about,[time(short)]).
adverb(afire,[action(burn)]).
adverb(after,[time(past)]).
adverb(ashore,[place(beach)]).
adverb(away,[direction(+)]).
adverb(crazily,[motion(eratic)]).
adverb(down,[vert_direction(-)]).
adverb(east,[direction(90)]).
adverb(fast,[motion(+)]).
adverb(frantically,[motion(+)]).
adverb(just,[time(0)]).
adverb(midway,[place(center)]).
adverb(north,[direction(0)]).
adverb(northeast,[direction(45)]).
adverb(northwest,[direction(315)]).
adverb(not,[state(-)]).
adverb(now,[time(0)]).
adverb(often,[frequency(high)]).
adverb(out,[direction(0)]).
adverb(quickly,[time(short)]).
adverb(sharply,[sharp(0)]).
adverb(soon,[time(short)]).
adverb(south,[direction(180)]).
adverb(southeast,[direction(135)]).
adverb(southwest,[direction(225)]).
adverb(still,[frequency(0)]).
adverb(today,[time(0)]).
adverb(tomorrow,[time(1)]).
adverb(up,[vert_direction(+)]).
adverb(well,[state(+)]).
adverb(west,[direction(270)]).
adverb(westward,[direction(270)]). /* bh */
adverb(wildly,[motion(eratic)]).
adverb(yesterday,[time(-1)]).

determiner(a,[indefinite]).
determiner(an,[indefinite]).
determiner(the,[definite]).

conjunction(and,[conjunction(and)]).
conjunction(as,[conjunction(as)]).
conjunction(but,[conjunction(and)]).
conjunction(or,[conjunction(or)]).
conjunction(plus,[conjunction(and)]).
conjunction(so,[conjunction(and)]).
conjunction(while,[conjunction(and)]).

infinmarker(to,[infinmarker]).

pronoun(another,[pronoun(another)]).
pronoun(several,[pronoun(several)]).
pronoun(that,[pronoun(that)]).

pronoun(he,[perprn(he)]).
pronoun(she,[perprn(she)]).
pronoun(it,[perprn(it)]).
pronoun(we,[perprn(we)]).
pronoun(they,[perprn(they)]).

pronoun(his,[posprn(his)]).
pronoun(her,[posprn(her)]).
pronoun(our,[posprn(our)]).
pronoun(their,[posprn(their)]).

clausehead(as,[clshead(as)]).
clausehead(that,[clshead(that)]).
clausehead(which,[clshead(which)]).
clausehead(who,[clshead(who)]).

:month('April',[name('April'),month_name]).
month('August',[name('August'),month_name]).
month('December',[name('December'),month_name]).
month('February',[name('February'),month_name]).
month('January',[name('January'),month_name]).
month('July',[name('July'),month_name]).
month('June',[name('June'),month_name]).
month('March',[name('March'),month_name]).
month('May',[name('May'),month_name]).
month('November',[name('November'),month_name]).
month('October',[name('October'),month_name]).
month('September',[name('September'),month_name]).
month('Apr',[name('April'),month_name]).
month('Aug',[name('August'),month_name]).
month('Dec',[name('December'),month_name]).
month('Feb',[name('February'),month_name]).
month('Jan',[name('January'),month_name]).
month('Jul',[name('July'),month_name]).
month('Jun',[name('June'),month_name]).
month('Mar',[name('March'),month_name]).
month('Nov',[name('November'),month_name]).
month('Oct',[name('October'),month_name]).
month('Sep',[name('September'),month_name]).

noun(action,[action]).
noun(afternoon,[time_loc]).
noun(air,[air]).
noun(aircraft,[aircraft]).
noun(aircraft_carrier,[ship,aircraft_carrier]).
noun(area,[region]).
noun(army,[army,organization]).
noun(arroyo,[river,geofeature]).
noun(atlantic,[ocean]).
noun(attack,[attack]).
noun(attention,[attention]).
noun(autumn,[time_loc]).
noun(base,[base]).
noun(based,[base]).
noun(battleship,[superclasses(battleship,[warship]),battleship,size(+)]).
noun(beach,[geofeature]).
noun(beaches,[geofeature]).
noun(bend,[turn]).
noun(blow,[blow]).
noun(boat,[ship]).
noun(bomb,[bomb]).
noun(bombardment,[bombardment]).
noun(bomber,[bomber]).
noun(boundary,[edge,boundary]).
noun(bow,[bow]).
noun(brush,[vegetation(-)]).
noun(bunch,[set]).
noun(bush,[plant(0)]).
noun(caliber,[caliber]).
noun(carrier,[carrier,vehicle]).
noun(cave,[geofeature]).
noun(class,[class]).
noun(column,[column]).
noun(convoy,[set]).
noun(couple,[set]).
noun(cover,[terrain]).
noun(craft,[craft]).
noun(creek,[river,geofeature]).
noun(crossroads,[junction]).
noun(cruiser,[superclasses(cruiser,[warship]),size(-)]).
noun(cutter,[ship,cutter]).
noun(dawn,[time_loc]).
noun(day,[time_loc]).
noun(decade,[time_loc]).
noun(demining,[demine]).
noun(destroyer,[superclasses(destroyer,[warship]),size(-)]).
noun(destruction,[destruction]).
noun(dusk,[time_loc]).
noun(earth,[terrain,earth]).
noun(east,[region,right_region,xcoordinate(+)]).
noun(edge,[edge,boundary]).
noun(effort,[effort]).
noun(enemy,[enemy]).
noun(engagement,[engagement]).
noun(evening,[time_loc]).
noun(fall,[time_loc]).
noun(fight,[fight]).
noun(firing,[fire]).
noun(flame,[flame]).
noun(fleet,[set]).
noun(force,[force]).
noun(forest,[vegetation(+)]).
noun(foreground,[foreground]).
noun(fregate,[ship,fregate]).
noun(freighter,[ship,freighter]).
noun(future,[time_loc]).
noun(grass,[terrain,grass]).
noun(ground,[ground]).
noun(group,[set]).
noun(gun,[gun]).
noun(gunboat,[ship,gunboat]).
noun(half,[half,region,size(big)]).
noun(harbor,[harbor]).
noun(hill,[hill,geofeature]).
noun(hour,[time_loc]).
noun(infantry,[infantry]).
noun(installation,[installation]).
noun(intersection,[junction]).
noun(invader,[invader]).
noun(invasion,[invation]).
noun(jeep,[jeep,vehicle]).
noun(junction,[junction]).
noun(lake,[sea(-)]).
noun(land,[terrain,earth]).
noun(landing,[landing]).
noun(left,[region,xcoordinate(-)]).
noun(life,[life]).
noun(line,[line]).   /* KMW */
noun(litter,[litter]).
noun(maneuver,[maneuver]).
noun(marine,[marine]).
noun(merchant,[merchant]).
noun(middle,[region,xcoordinate(0),ycoordinate(0)]).
noun(midnight,[time_loc]).
noun(minute,[time_loc]).
noun(month,[time_loc]).
noun(morning,[time_loc]).
noun(motion,[motion]).
noun(mountain,[mountain,geofeature]).
noun(moving,[move]).
noun(navy,[navy,organization]).
noun(night,[time_loc]).
noun(noon,[time_loc]).
noun(north,[region,upper_region,ycoordinate(+)]).
noun(northeast,[region,upper_right_region,xcoordinate(+),ycoordinate(+)]).
noun(northwest,[region,upper_left_region,xcoordinate(-),ycoordinate(+)]).
noun(number,[set]).
noun(object,[region]).
noun(ocean,[sea,geofeature]).
noun(order,[order]).
noun(past,[time_loc]).
noun(path,[road(-)]).
noun(plane,[plane]).
noun(plant,[plant]).
noun(pond,[lake(-)]).
noun(position,[position]).
noun(present,[time_loc]).
noun(raid,[action]).
noun(region,[region]).
noun(resistance,[resistance]).
noun(right,[region,xcoordinate(+)]).
noun(river,[river,geofeature]).
noun(road,[road(+)]).
noun(rock,[terrain,rock]).
noun(route,[road(+)]).
noun(sand,[terrain,sand]).   /* KMW */
noun(sea,[sea,geofeature]).
noun(second,[time_loc]).
noun(set,[set]).
noun(shape,[region]).
noun(shell,[shell]).
noun(ship,[ship,vehicle]).
noun(shore,[shore]).
noun(shot,[shot]).
noun(shrub,[plant(0)]).
noun(side,[side,region]).
noun(sight,[sight]).
noun(smoke,[smoke]).
noun(soldier,[soldier]).
noun(south,[region,lower_region,ycoordinate(-)]).
noun(southeast,[region,lower_right_region,xcoordinate(+),ycoordinate(-)]).
noun(southwest,[region,lower_left_region,xcoordinate(-),ycoordinate(-)]).
noun(spot,[region]).
noun(spring,[time_loc]).
noun(steam,[steam]).
noun(stern,[stern]).
noun(stream,[river,geofeature]).
noun(stretch,[region]).
noun(strike,[strike]).
noun(strip,[strip,region,shape(narrow)]).
noun(stronghold,[stronghold]).
noun(submarine,[ship,under_surface_vehicle]).
noun(summer,[time_loc]).
noun(sunrise,[time_loc]).
noun(sunset,[time_loc]).
noun(tank,[tank,vehicle]).
noun(tanker,[ship,tanker]).
noun(terrain,[terrain]).
noun(there,[location]).
noun(ton,[ton]).
noun(track,[road(-)]).
noun(tree,[plant(+)]).
noun(trouble,[trouble]).
noun(turn,[turn]).
noun(u-boat,[ship,under_surface_vehicle]).
noun(vain,[vain]).
noun(vegetation,[vegetation]).
noun(vessel,[ship,vehicle]).
noun(victory,[victory]).
noun(waiting,[wait,action]).
noun(warehouse,[warehouse]).
noun(warship,[superclasses(warship,[navy,ship]),warship,color(gray)]).
noun(water,[water]).
noun(wave,[sea,geofeature]).
noun(way,[way]).
noun(week,[time_loc]).
noun(west,[region,left_region,xcoordinate(-)]).
noun(winter,[time_loc]).
noun(year,[time_loc]).
```

```
preposition(after,[property(time_spec)]).
preposition(along,[property(location)]).
preposition(among,[property(part_inside)]).
preposition(as,[property(coagent)]).
preposition(at,[property(location)]).
preposition(at,[property(time_spec)]).
preposition(before,[property(time_spec)]).
preposition(below,[property(location)]).
preposition(beside,[property(location)]).
preposition(between,[property(location)]).
preposition(by,[property(coagent)]).
preposition(by,[property(location)]).
preposition(during,[property(time_spec)]).
preposition(for,[property(beneficiary)]).
preposition(for,[property(purpose)]).
preposition(for,[property(location)]).
preposition(for,[property(time_spec)]).
preposition(from,[property(source)]).
preposition(from,[property(time_src)]).
preposition(in,[property(inside)]).
preposition(in,[property(in_period)]).
preposition(into,[property(inside)]).
preposition(near,[property(location)]).
preposition(of,[property(coagent)]).
preposition(of,[property(part_of)]).
preposition(of,[property(subtype)]).
preposition(off,[property(location)]).
preposition(on,[property(location)]).
preposition(on,[property(time_spec)]).
preposition(onto,[property(above)]).
preposition(over,[property(above)]).
preposition(through,[property(part_inside)]).
preposition(through,[property(time_spec)]).
preposition(to,[property(destination)]).
preposition(to,[property(time_dest)]).
preposition(under,[property(below)]).
preposition(until,[property(time_spec)]).
preposition(with,[property(coagent)]).
preposition(with,[property(contains)]).
preposition(with,[property(tool)]).
preposition(within,[property(in_period)]).
preposition(within,[property(inside)]).
```

```
propnoun('carrier-based',[name('carrier-based'),coagent]).
propnoun('land-locked',[name('land-locked'),geofeature]).
propnoun('pre-invasion',[name('pre-invation'),action]).

propnoun('Hunter-Liggett',[name('Hunter-Liggett'),place]).
propnoun('Christmas',[name('Christmas'),holiday_name]).
propnoun('Japanese',[name('Japanese')]).
propnoun('Jolon',[name('Jolon'),place]).
propnoun('Kongo',[name('Kongo')]).
propnoun('LCI',[name('LCI(G)')]).
propnoun('MacArthur',[name('MacAthur')]).
propnoun('Macy''s',[name('Macy''s'),place]).
propnoun('Marine',[name('Marine'),military]).
propnoun('Marines',[name('Marines'),military]).
propnoun('Marine','Corps',[name('Marine Corps'),military]).
propnoun('Midway',[name('Midway'),place]).
propnoun('Missouri',[name('Missouri')]).
propnoun('Morotai',[name('Morotai'),place]).
propnoun('Nachi',[name('Nachi')]).
propnoun('Navy',[name('Navy')]).
propnoun('Nacimiento',[name('Nacimiento'),place]).
propnoun('Pacific',[name('Pacific Ocean'),place]).
propnoun('Philippines',[name('Philippines'),place]).
propnoun('Rabaul',[name('Rabaul'),place]).
propnoun('Saratoga',[name('Saratoga')]).
propnoun('Shokaku',[name('Shokaku')]).
propnoun('Soryu',[name('Soryu')]).
propnoun(['U',period,'S',period],[name('U.S.'),place]).
propnoun(['Columbus','Day'],[name('Columbus Day'),holiday_name]).
propnoun(['Independence','Day'],[name('Independence Day'),holiday_name]).
propnoun(['Fifth','Amphibious','Corps'],[name('Fifth Amphibious Corps')]).
propnoun(['Iwo','Jima'],[name('Iwo Jima'),place]).
propnoun(['Landing','Craft','Infantry','Gunboats'],[name('LCI(G)')]).
propnoun(['Manila','Bay'],[name('Manila Bay'),place]).
propnoun(['Morotai','Island'],[name('Morotai Island'),place]).
propnoun(['New','Britain','Island'],[name('New Britain Island'),place]).
propnoun(['New','Britain'],[name('New Britain'),place]).
propnoun(['New','Guinea'],[name('New Guinea'),place]).
propnoun(['Pacific','Ocean'],[name('Pacific Ocean'),place]).
propnoun(['Philippine','Sea'],[name('Philippine Sea'),place]).
propnoun(['Rabaul','Harbor'],[name('Rabaul Harbor'),place]).
propnoun(['Sagami','Bay'],[name('Sagami Bay'),place]).
propnoun(['Task','Force'],[name('Task Force')]).
propnoun(['Tokyo','Bay'],[name('Tokyo Bay'),place]).
propnoun(['World','War','II'],[name('World War II'),war(ww2)]).
propnoun('Zeros',[name('Zeros')]).

aux(being,[tense(past),singular]).
aux(can,[tense(present),singular]).
aux(could,[tense(past),singular]).
aux(had,[tense(past),singular]).
aux(has,[tense(past),singular]).
aux(have,[tense(past),plural]).
aux(may,[possibility]).
aux(will,[tense(future)]).

doesword(did,[tense(past)]).
doesword(do,[tense(present)]).
doesword(does,[tense(present)]).
doesword(has,[tense(past),singular]).
doesword(have,[tense(past),plural]).

tobe(is,[tense(present)]).
tobe(was,[tense(past)]).
tobe(were,[tense(past),plural]).
tobe(were,[tense(subjunctive),singular]).

verb(abandon,[transitive]).
verb(arrive,[intransitive]).
verb(attack,[transitive]).
verb(avoid,[transitive]).
verb(base,[transitive]).
verb(batter,[transitive]).
verb(bend,[transitive]).
verb(blow,[transitive]).
verb(bomb,[transitive]).
verb(burn,[transitive]).
verb(burst,[transitive]).
verb(bursting,[transitive]).
verb(camouflage,[transitive]).
verb(can,[transitive]).
verb(churn,[transitive]).
verb(circle,[transitive]).
verb(clear,[transitive]).
verb(come,[transitive]).
verb(cross,[transitive]).
verb(cut,[transitive]).
verb(defend,[transitive]).
verb(demine,[transitive]).
verb(depart,[intransitive]).
verb(destroy,[transitive]).
verb(dive,[transitive]).
verb(end,[transitive]).
verb(escape,[transitive]).
verb(evade,[transitive]).
verb(fight,[transitive]).
verb(fire,[transitive]).
verb(flank,[transitive]).
verb(force,[transitive]).
verb(get,[transitive]).
verb(go,[transitive]).
verb(gyrate,[intransitive]).
verb(have,[transitive]).
verb(head,[transitive]).
verb(hit,[transitive]).
verb(inflict,[transitive]).
verb(join,[transitive]).
verb(knock,[transitive]).
verb(know,[transitive]).
verb(land,[transitive]).
verb(landed,[transitive]).
verb(lead,[transitive]).
verb(leave,[transitive]).
verb(litter,[transitive]).
verb(lock,[transitive]).
verb(make,[transitive]).
verb(maneuver,[transitive]).
verb(meet,[transitive]).
verb(move,[motion,transitive]).
verb(neutralize,[transitive]).
verb(open,[transitive]).
verb(operate,[transitive]).
verb(order,[transitive]).
verb(own,[transitive]).
verb(pour,[transitive]).
verb(present,[transitive]).
verb(protect,[transitive]).
verb(raid,[transitive]).
verb(reach,[transitive]).
verb(reinforce,[transitive]).
verb(remain,[intransitive]).
verb(rise,[intransitive]).
verb(run,[transitive]).
verb(sail,[motion,transitive]).
verb(score,[transitive]).
verb(scurry,[intransitive]).
verb(see,[transitive]).
verb(separate,[transitive]).
verb(settle,[transitive]).
verb(set,[transitive]).
verb(shape,[transitive]).
verb(shoot,[transitive]).
verb(sink,[transitive]).
verb(smash,[transitive]).
verb(smoke,[transitive]).
verb(stay,[transitive]).
verb(steam,[transitive]).
verb(storm,[transitive]).
verb(strike,[transitive]).
verb(surround,[transitive]).
verb(take,[transitive]).
verb(terminate,[transitive]).
verb(try,[transitive]).
verb(turn,[transitive]).
verb(view,[transitive]).
verb(visit,[transitive]).
verb(wade,[transitive]).
verb(wait,[transitive]).
verb(want,[transitive]).
```

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Commandant of the Marine Corps Code TE 06 Headquarters, U.S. Marine Corps Washington, D.C. 20380-0001	1
4.	Department Chairman, Code 52 Department of Computer Science Naval Postgraduate School Monterey, CA 93943-5100	1
5.	Professor Vincent Y. Lum, Code 52Lm Department of Computer Science Naval Postgraduate School Monterey, CA 93943-5100	2
6.	Associate Professor Neil C. Rowe, Code 52Rp Department of Computer Science Naval Postgraduate School Monterey, CA 93943-5100	2
7.	Capt. John D. Dulle 510 Devonshire Dr., N. E. Vienna, VA 22180	2